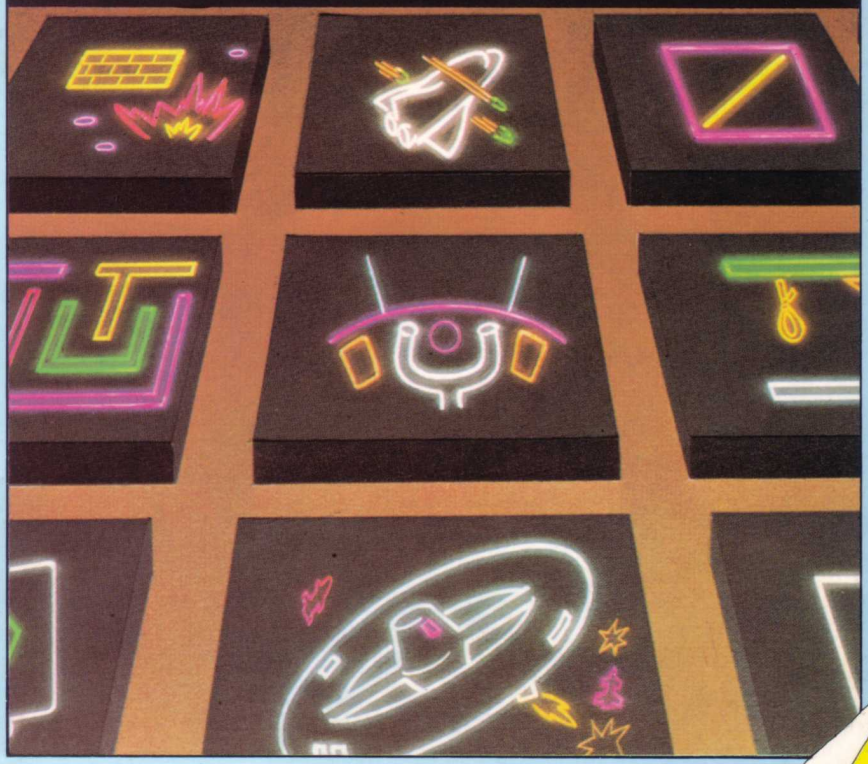
Computer Games Series

ALTRI

GIOCHI PER IL VOSTRO ATARI



GREMESE EDITORE

Virgin

Computer Games Series

ALTRI GIOCHI PER IL VOSTRO ATARI

Computer Games Series
Periodico mensile
N. 12 - Febbraio 1985
Registrazione Tribunale di Roma N. 138/84
del 24 marzo 1984
Direttore Responsabile: Grazia Valci

Volumi pubblicati nella collana:

Giochiamo con COMMODORE 64
Giochiamo con VIC 20
Giochiamo con ZX 81
Giochiamo con ZX SPECTRUM
Giochiamo con ATARI
Giochiamo con TRS-80
Altri giochi per il vostro COMMODORE 64
Altri giochi per il vostro VIC 20
Altri giochi per il vostro ZX SPECTRUM
Giochiamo con TI 99/4A
Altri giochi per il vostro ZX 81
Altri giochi per il vostro ATARI

Computer Games Series

ALTRI GIOCHI PER IL VOSTRO ATARI

Gary Ryan
e
Cliff McConnell

GREMESE EDITORE

Titolo originale
More games for your ATARI 600 XL

Traduzione dall'inglese
Paolo Musu

Edizione italiana a cura di
Giancarlo Zagarese

Design
Ray Hyden

Illustrazioni
Sue Walliker

Fotocomposizione
Linotipia De Angelis - Roma

Stampa
Arti Grafiche LABOR - Todi

© 1984 Interface/Virgin Books
© 1985 GREMESE EDITORE s.r.l.
Via Virginia Agnelli, 88 - 00151 Roma

Tutti i diritti riservati. Nessuna parte di questo libro può essere riprodotta, registrata o trasmessa, in qualsiasi modo e con qualsiasi mezzo, senza il preventivo consenso dell'Editore.

ISBN 88-7605-153-8

INDICE

Introduzione	7
Introduzione dell'autore	8
Introduzione del curatore italiano	9
Note sui programmi	10
Traffic Panic (Traffico da panico)	11
Breakthrough	19
Shootout (Bombardamento)	27
Tower of Flames (Grattacielo in fiamme)	31
Asteroid Storm (Tempesta di asteroidi)	38
Blockout	45
Space Attack (Attacco dallo spazio)	49
Gobbler (Tacchino)	57
Grand Prix	64
Escape (Fuga)	68
Time Typist (Dattilografo lampo)	71
Galax Attack (Attacco della Galassia)	73
Chopper Mission (Missione di salvataggio)	85
Reversi	96
Mars Lander (Atterraggio su Marte)	101
Come scrivere programmi migliori	106
Glossario	111
Traduzioni	125

GARY RYAN e CLIFF McCONNELL — AUTORI

Gary Ryan e Cliff McConnell sono due studenti diciottenni di Canberra, capitale dell'Australia, particolarmente versati per il gioco, lo sport e la bella vita, alla quale si dedicano con molto piacere, ma solo nei pochi momenti che riescono a sottrarre allo studio e al rapporto odio-amore con il computer.

TIM HARTNELL
CURATORE DELL'EDIZIONE INGLESE

Tim Hartnell è un eminente giornalista la cui esperienza computeristica ha contribuito a determinare il successo della Technical Consumer Press. È anche autore di diversi libri, tra i quali: *Getting Acquainted With Your ZX81, Let Your BBC Micro Teach You to Program* e *Programming Your ZX Spectrum.*

GIANCARLO ZAGARESE
CURATORE DELL'EDIZIONE ITALIANA

Giancarlo Zagarese, insegnante di discipline scientifiche, è autore di oltre 250 articoli e di vari volumi nei settori dell'elettronica e dell'attività subacquea. Per GREMESE EDITORE ha già collaborato in *Il Sub per tutti* e *Il Sub in apnea* della serie "gli Abbicci".

SUE WALLIKER — ILLUSTRATRICE

Sue Walliker è un'illustratrice free-lance.

Introduzione

Il vostro computer è in attesa per sfidarvi. Rapidi «games» di grafica, giochi di concentrazione, di parole e di enigmistica sono tutti qui pronti a farvi divertire.
Nel libro vi sono una notevole varietà di giochi i cui programmi sono stati scritti da alcuni fra i più giovani e abili programmatori che lavorano attualmente in Gran Bretagna. L'esaminare i programmi dei giochi vi dà la possibilità di apprendere raffinate tecniche e sottili metodi di programmazione che voi stessi potete poi applicare. Inoltre, una volta che avete conosciuto a fondo i programmi presentati dal libro, potreste senz'altro provare a migliorarli — un programma non è mai "perfetto" — arricchendo le vostre capacità di programmazione. Adesso voltate pagina e incominciate a "battere" i programmi. Sono certo che il vostro divertimento sarà pari a quello che abbiamo provato durante la stesura di questo volume.

Introduzione dell'autore

Dopo diversi mesi dedicati all'ideazione,alla progettazione e ad una accurata programmazione, vi presentiamo finalmente il libro che attendevate. Ognuno di noi ha preferenze per un particolare tipo di gioco, abbiamo perciò cercato di soddisfare tutte le esigenze: abbiamo inserito giochi di carte, di strategia, di simulazione e, soprattutto, i veloci arcade games. A differenza di altri "listati", tutti i nostri giochi sono efficientemente programmati, veloci, colorati e divertenti da giocare.
Tutti i giochi di questo volume sono stati concepiti per il nuovissimo ATARI 600 XL, ma sono tutti compatibili sia per il 400 sia per l'800.
Non vi rimane che sedervi al computer, inserire i programmi e... buon divertimento!

Gary Ryan e Cliff McConnell
Canberra, Australia
Marzo 1984

Prefazione all'edizione italiana

È con piacere che ho accolto l'invito dell'editore Gremese per curare una serie, finalmente in italiano, di volumetti sui videogiochi. Molti acquirenti di piccoli e medi personal computer, sia che lo utilizzino personalmente sia che ne abbiamo fatto oggetto di regalo ai propri figli, si sono fatti trascinare dalla pubblicità che precisava «al prezzo di un semplice video-gioco acquistate un intero computer». È vero ed è stato un buon acquisto. Per imparare ad utilizzare un computer in modo semplice e divertente non c'è però niente di meglio della via ludica. Non giochi comprati e fruiti passivamente, però, ma "creati" e vissuti, istruzione dopo istruzione, prima compiendo e poi personalizzando o "inventando", in modo da accedere gradualmente nel nuovo mondo dell'informatica.

g.z.

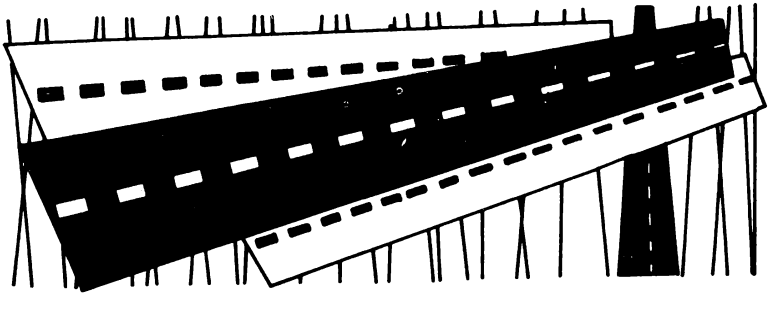
Note sulla programmazione

Nel battere questi programmi dovete essere a conoscenza di alcuni simboli.
Trovando il simbolo "I" dovete premere il tasto INVERSE.
Il simbolo "↑" indica che dovete tener premuto il tasto CONTROL che verrà rilasciato quando appare "←".
Per sistemi che usano un DOS diverso dal DOS ATARI (DOS 2), può essere necessaria maggior memoria di quella dichiarata. Questo solo se il DOS in uso prende più memoria. ("Asteroid Storm", per esempio, non girerà con un DOS che occupi più memoria del DOS 2 dell'Atari. Qualche altro gioco presentato — quelli che adoperano il codice macchina — non girerà subito con una versione DOS diversa dall'Atari. Se scoprite che un programma non funziona con il vostro DOS, caricate il programma, battete POKE 9,0 e premete RETURN; premete infine SYSTEM RESET per giocare).
Come vedrete, ogni gioco del libro fa uso di alcune, o di tutte, le seguenti pregiate caratteristiche dell'Atari: grafica ad alta definizione, controllo definizione caratteri, suono, controlli grafici e del colore, ecc.
Molti programmi adoperano il linguaggio macchina e diversi ne fanno comunque un ampio uso.

TRAFFIC PANIC

(CASSETTA 16K, DISCO 24K)

Siamo nel 2039 e, a causa dell'affollamento delle strade e della mancanza di spazio, le nuove vie di comunicazione sono state costruite una sopra l'altra. Vi trovate nella più bassa di queste autostrade e vi piacerebbe godervi la vista dall'alto: per togliervi questa sodddisfazione dovete arrampicarvi per cinque livelli adoperando delle scale abbandonate; ogni livello è però affollato di traffico, e l'unico modo per non finire "arrotati" è di saltare a bordo delle auto in arrivo.
Dopo che il programma avrà inizializzato il linguaggio macchina il titolo apparirà sullo schermo. Premete SELECT per cambiare lo stadio da dove iniziate. Premete START per cominciare il gioco.
Avete tre "vite" a disposizione: dopo averle perdute tutte il programma tornerà alla videata iniziale indicando il punteggio più alto. Per salire una scala (se già ci siete o vi ci trovate sotto) spostate la levetta del joystick verso l'alto; per scendere spostatela verso il basso, per andare a destra o a sinistra spostate la levetta a destra o a sinistra. Per saltare premete il pulsante FIRE; per saltare verso destra o verso sinistra operate contemporaneamente premendo sul pulsante e spostando la leva del joystick verso destra o verso sinistra.



```
10 REM **TRAFFIC PANIC
20 REM **Written by Cliff McConnell
30 REM **ALL REMS MAY BE OMITTED
60 DIM S(5),N(5),X1(5),Y1(5),X2(5),Y2(
5)
110 TOP=PEEK(740):PMBASE=(TOP-8)*256:S
TART=(TOP-12)*256:POKE 106,TOP:POKE 0,
PMBASE/256:GRAPHICS 0
115 POKE 106,TOP-3:GRAPHICS 0:POKE 106
,TOP-5:CHSET=(TOP-14)*256
120 VBLANK=START:DLI=START+604:USER=ST
ART+645
130 FOR K=0 TO 5:X1(K)=0:Y1(K)=0:X2(K)
=0:Y2(K)=0:NEXT K
140 GRAPHICS 17
150 POSITION 6,5:? #6;"TRAFFIC":POSITI
ON 7,7:? #6;"panic"
210 GOTO 510
300 REM PAGE 6 DATA
310 DATA 0,0,0,0,28,28,62,62,255,255,2
55,66,0,0,0,0,1,130,3,132,5,134,133,15
3,173,193
311 DATA 248,12,32,52,142,162,182,202,
239,3,23,43,151,171,191,211,230,250,14
,34,11,15,2,8,7,14
312 DATA 0,0,0,24,24,60,60,126,66,66,1
26,60,60,24,24,0,0,0,0,24,24,60,36,102
,102,102
313 DATA 102,36,60,24,24,0,0,0,0,24,24
,60,60,126,66,66,126,60,60,24,24,0,0,2
4,24,60
314 DATA 60,126,66,66,126,60,60,24,24,
0,0,0,100,-1
400 REM MACHINE LANGUAGE
410 DATA 169,4,141,128,6,162,3,189,22,
6,157,0,208,202,16,247,173,139,6,240,3
,76,95,228,198,207
411 DATA 208,53,165,208,133,207,160,18
,177,88,201,16,208,6,169,25,145,88,208
,27,56,233,1,145,88,201
```

```
412 DATA 16,208,26,136,177,88,201,16,2
08,19,169,1,141,139,6,141,140,6,76,95,
228,136,177,88,56,233
413 DATA 1,145,88,162,5,189,16,6,41,11
2,74,74,74,74,133,203,189,16,6,41,15,2
4,125,46,6,157
414 DATA 46,6,201,16,144,7,41,15,157,4
6,6,230,203,138,10,10,168,24,105,4,133
,204,189,16,6,48
415 DATA 16,185,22,6,24,101,203,153,22
,6,200,196,204,208,242,240,14,185,22,6
,56,229,203,153,22,6
416 DATA 200,196,204,208,242,202,16,17
5,173,131,6,208,127,173,137,6,208,31,1
73,16,208,208,84,169,1,141
417 DATA 138,6,173,0,211,106,106,106,1
76,5,162,0,142,138,6,106,176,5,162,2,1
42,138,6,238,137,6
418 DATA 173,137,6,201,21,176,5,206,13
3,6,208,12,238,133,6,201,40,208,5,169,
0,141,137,6,173,132
419 DATA 6,24,109,138,6,56,233,1,141,1
32,6,201,48,176,3,238,132,6,201,210,14
4,3,206,132,6,169
420 DATA 1,208,63,173,0,211,106,106,10
6,176,27,206,132,6,173,132,6,201,48,17
6,3,238,132,6,206,130
421 DATA 6,208,7,169,3,141,130,6,208,3
2,208,38,106,176,35,238,132,6,173,132,
6,201,210,144,3,206
422 DATA 132,6,206,130,6,208,17,169,3,
141,130,6,208,2,208,81,173,135,6,73,1,
141,135,6,173,0
423 DATA 211,106,176,83,173,131,6,208,
33,174,129,6,189,116,6,205,132,6,208,1
2,169,0,141,134,6,169
424 DATA 1,141,131,6,208,10,189,122,6,
205,132,6,208,47,240,234,206,133,6,238
,134,6,206,136,6,208
425 DATA 13,169,4,141,136,6,173,135,6,
```

```
73,1,141,135,6,173,134,6,201,32,208,10
0,169,0,141,131,6
426 DATA 141,135,6,238,129,6,24,144,86
,106,176,83,173,131,6,208,44,174,129,6
,240,73,202,189,116,6
427 DATA 205,132,6,208,20,142,129,6,16
9,32,141,134,6,169,2,141,135,6,169,1,1
41,131,6,208,10,189
428 DATA 122,6,205,132,6,208,36,240,22
6,206,136,6,208,13,169,4,141,136,6,173
,135,6,73,1,141,135
429 DATA 6,238,133,6,206,134,6,208,8,1
69,0,141,131,6,141,135,6,173,135,6,10,
10,10,10,170,165
430 DATA 0,24,105,3,133,204,169,0,133,
203,169,15,133,205,172,133,6,189,52,6,
145,203,200,232,198,205
431 DATA 16,245,173,132,6,141,4,208,56
,233,2,141,5,208,56,233,2,141,6,208,56
,233,2,141,7,208
432 DATA 162,0,173,133,6,205,141,6,240
,4,141,141,6,170,142,0,210,232,142,2,2
10,169,168,141,1,210
433 DATA 141,3,210,76,95,228,72,138,72
,174,128,6,189,22,6,141,0,208,189,23,6
,141,1,208,189,24
434 DATA 6,141,2,208,189,25,6,141,3,20
8,232,232,232,232,142,128,6,104,170,10
4,64,104,169,0,141,139
435 DATA 6,173,133,6,201,34,176,6,169,
1,141,139,6,96,173,9,208,24,109,10,208
,24,109,140,6,240
436 DATA 230,141,139,6-96,0,-1
440 REM CHARACTER SET DATA
442 DATA 255,165,255,0,0,0,0,0
444 DATA 99,127,127,99,99,127,127,99
446 DATA 255,165,255,99,99,127,127,99
448 DATA 0,12,30,63,63,30,12,0
450 REM DATA FOR CARS
460 DATA 28,28,62,62,255,255,255,66
```

```
470 DATA 56,56,124,124,255,255,255,66
510 TRAP 1020
520 REM READ IN PAGE 6 DATA
1010 FOR T=1536 TO 9999:READ A:POKE T,
A:NEXT T
1020 TRAP 1520
1030 REM READ IN ML
1510 FOR T=START TO START+9999:READ A:
POKE T,A:NEXT T
1520 POKE 54279,PMBASE/256:POKE 53277,
3:POKE 559,62:POP
1524 FOR T=0 TO 511:POKE CHSET+T,PEEK(
57344+T):NEXT T
1527 FOR T=0 TO 31:READ A:POKE CHSET+2
4+T,A:NEXT T
1530 FOR K=40 TO 227 STEP 64:RESTORE 4
60:FOR T=0 TO 7:READ A:POKE PMBASE+102
4+K+T,A:POKE PMBASE+1280+K+T,A
1540 POKE PMBASE+1536+K+T,A:POKE PMBAS
E+1792+K+T,A:NEXT T:NEXT K
1550 FOR K=72 TO 227 STEP 64:RESTORE 4
70:FOR T=0 TO 7:READ A:POKE PMBASE+102
4+K+T,A:POKE PMBASE+1280+K+T,A
1560 POKE PMBASE+1536+K+T,A:POKE PMBAS
E+1792+K+T,A:NEXT T:NEXT K
3010 POKE 704,202:POKE 705,122:POKE 70
6,74:POKE 707,250:POKE 623,16:POKE 708
,108:POKE 709,172:POKE 710,150
3015 POKE 711,54
3020 DL=PEEK(560)+256*PEEK(561):FOR K=
8 TO 24 STEP 4:POKE DL+K,PEEK(DL+K)+12
8:NEXT K:POKE 756,CHSET/256
3030 FOR T=2 TO 22 STEP 4:POSITION 0,T
:? #6;"######################":NEXT T
3040 HI=INT(VBLANK/256):POKE 54286,0:P
OKE 547,HI:POKE 546,VBLANK-256*HI:HI=I
NT(DLI/256):POKE 513,HI
3050 POKE 512,DLI-256*HI:SC=0:HS=SC:LE
VEL=1:POKE 54281,CHSET/256
4000 REM WAIT FOR START/SELECT
```

```
4010 POP :POP :POSITION 1,3:? #6;"pres
s start/select"
4015 POSITION 6,5:? #6;"TRAFFIC":POSIT
ION 7,7:? #6;"panic"
4020 POSITION 0,0:? #6;"score:";SC
4022 POSITION 0,1:? #6;"high score:";H
S
4025 POSITION 12,0:? #6;"level";LEVEL;
"  "
4030 IF PEEK(53279)=6 THEN 7010
4035 IF PEEK(53264)=0 THEN 7010
4040 IF PEEK(53279)<>5 THEN 4030
4050 LEVEL=LEVEL*(LEVEL<9)+1:POSITION
17,0:? #6;LEVEL:POSITION 13,3:? #6;"SE
LECT"
4060 IF PEEK(53279)=5 THEN 4060
4070 POSITION 13,3:? #6;"select":GOTO
4030
7000 REM PRESSED START
7010 POSITION 7,3:? #6;"START"
7020 IF PEEK(53279)=6 THEN 7020
7030 POSITION 1,3:? #6;"
    ":SC=0:POSITION 6,0:? #6;SC;"     "
:POSITION 12,0:? #6;"       "
7110 STAGE=LEVEL:MEN=3:POSITION 6,5:?
#6;"        "
7115 POSITION 6,7:? #6;"STAGE ";STAGE:
POKE 77,0
7120 FOR K=0 TO 5:S(K)=8+STAGE*2.5+5-K
+128*(INT(K/2)=K/2)
7130 NEXT K
7135 FOR K=0 TO 5
7140 IF Y1(K)=0 THEN 7150
7145 POSITION X1(K),Y1(K):? #6;"#":FOR
 T=1 TO 3:IF Y1(K)+T>23 THEN 7148
7146 LOCATE X1(K),Y1(K)+T,A:IF A<39 TH
EN POSITION X1(K),Y1(K)+T:? #6;" "
7148 NEXT T
7150 IF Y2(K)=0 THEN 7160
7155 POSITION X2(K),Y2(K):? #6;"#":FOR
```

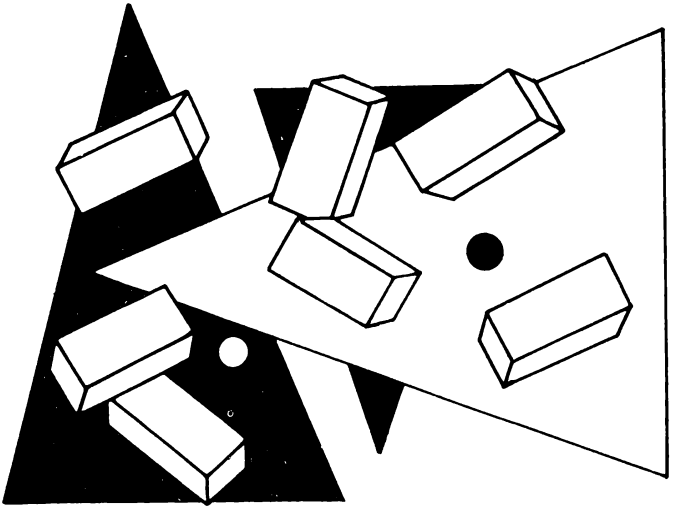
```
 T=1 TO 3:IF Y2(K)+T>23 THEN 7158
7156 LOCATE X2(K),Y2(K)+T,A:IF A<39 TH
EN POSITION X2(K),Y2(K)+T:? #6;" "
7158 NEXT T
7160 NEXT K
7170 FOR K=0 TO 5:N(K)=STAGE*(STAGE<5)
+4*(STAGE>4)-1:NEXT K
7190 FOR K=0 TO 23:POKE 1558+K,0:NEXT
K
7210 FOR T=0 TO 5:N=0:FOR K=0 TO N(T):
POKE 1558+T*4+K,N:N=N+25+INT(RND(0)*39
):NEXT K:NEXT T
7310 FOR K=0 TO 5:POKE 1552+K,S(K):NEX
T K
7320 POP :POP
7510 FOR L=0 TO 5:HPOS=56+8*INT(RND(0)
*18):GOSUB 20010
7520 POKE 1652+5-L,HPOS-1:X1(L)=X:Y1(L
)=Y:NEXT L
7530 FOR L=0 TO 5:POKE 1658+5-L,0:IF R
ND(0)<0.3 THEN HPOS=56+8*INT(RND(0)*18
):GOSUB 20010:POKE 1658+5-L,HPOS-1
7540 X2(L)=X:Y2(L)=Y:NEXT L
8030 FOR X=6 TO 13:LOCATE X,7,A:IF A>3
8 THEN A=32
8040 POSITION X,7:PUT #6,A:NEXT X
9030 FU=500:POSITION 12,0:? #6;"time "
;FU:POKE 207,50:POKE 208,50
10010 FOR T=PEEK(1669) TO PEEK(1669)+1
4:POKE PMBASE+768+T,0:NEXT T
10015 POKE 1668,100:POKE 1669,225
10020 POKE 1673,0:POKE 1665,0:POKE 166
7,0:POKE 1676,0
10030 POSITION 17,0:? #6;FU:COLOR 134
10060 FOR K=1 TO MEN:PLOT 16+K,1:NEXT
K
10070 IF MEN<3 THEN POSITION 17+MEN,1:
? #6;" ";
11010 POKE 1675,1:POKE 53278,0:POKE 54
```

```
286,192:A=USR(USER):SOUND 0,0,0,0:SOUN
D 1,0,0,0
11020 IF PEEK(53257)+PEEK(53258)+PEEK(
1676)>0 THEN 13010
12000 REM **END OF STAGE
12510 STAGE=STAGE+1:K=PEEK(88)+256*PEE
K(89):K=(PEEK(K+17)-16)*10+PEEK(K+18)-
16:K=K*10
12530 FOR FU=K TO 0 STEP -10:SC=SC+10*
STAGE:POSITION 6,0:? #6;SC:POSITION 17
,0:? #6;FU;:IF FU<100 THEN ? #6;" ";
12550 SOUND 0,70,10,10
12560 FOR K=0 TO 5:NEXT K:SOUND 0,0,0,
0:NEXT FU
12570 POSITION 17,0:? #6;" ";
12610 GOTO 7115
13000 REM **PLAYER DIED
13010 MEN=MEN-1:POSITION 17+MEN,1:? #6
;" "
13020 FOR T=0 TO 15:SOUND 0,255-T*17,1
0,10:POKE PMBASE+768+PEEK(1669)+T,0:FO
R K=0 TO 3:NEXT K:NEXT T
13100 IF MEN>0 THEN 10010
13110 POSITION 5,12:? #6;"game over":I
F SC>HS THEN HS=SC
13120 FOR K=1 TO 150:NEXT K:POSITION 5
,12:? #6;"          ":GOTO 4010
20000 REM
20010 X=(HPOS-56)/8:Y=2+(L*4)
20020 POSITION X,Y:? #6;"%":FOR T=1 TO
 3:IF Y+T<24 THEN POSITION X,Y+T:? #6;
"$"
20030 NEXT T
20040 RETURN
32000 GOTO 32000
```

BREAKTHROUGH

(CASSETTA 16K, DISCO 24K)

È la versione per computer del vecchio gioco in cui un muro multicolore viene abbattuto, mattone dopo mattone, con una palla fatta rimbalzare da una racchetta. Quando il muro sarà stato completamente distrutto, la racchetta verrà rimpicciolita e sarà eretto un nuovo muro. In questa versione la palla potrà rimbalzare secondo 16 diversi angoli.
Premete SELECT per modificare il numero di palle a disposizione.
Premete OPTION per cambiare il Breakthrough quando la palla, invece di rimbalzare via dal muro, va a crearsi un varco. Premete " f " o " s " per accelerare o rallentare la velocità della palla. Premete START per cominciare il gioco.
Usate uno joystick nella " port 1 " per muovere a destra o sinistra e premete il pulsante del fuoco per rallentare il movimento della vostra racchetta.



```
0 REM *** BREAKTHRU, BY G. RYAN
10 DIM A$(5):GOTO 50
20 RESTORE D:TRAP 40
30 FOR X=A TO A+400:READ D:POKE X,D:NE
XT X
40 POP :RETURN
50 RESTORE 50:READ N,N1,N2,N3,N4,N5,N6
,N7,N8,N9,N10,N11,N12,N16,N20,F,W,KB,C
SL:DATA 0,1,2,3,4,5,6
55 DATA 7,8,9,10,11,12,16,20,255,256,7
64,53279
59 REM SETUP TOP OF RAM
60 R=INT(PEEK(740)/N4)*N4:POKE 106,R:G
RAPHICS N20+N2:POKE 106,R-N8:RT=R*W
70 VBI=RT-180:VBD=RT-360:PLR=RT-512:MS
L=RT-640:MAIN=RT-960:DLI=RT-1080:PMB=R
-N4:INIT=RT-1024:DL=RT-1150
80 GRAPHICS N2+N16:GOSUB 4500:POSITION
 N6,N5:? #6;"breakthru":POSITION N1,N7
:? #N6;"WRITTEN BY G. RYAN"
90 D=5500:A=VBI:GOSUB N20:D=6500:A=VBD
:GOSUB N20
100 D=5000:A=MAIN:GOSUB N20:D=7500:A=D
LI:GOSUB N20
110 D=6000:A=1024:GOSUB N20:D=8000:A=1
760:GOSUB N20:D=7000:A=INIT:GOSUB N20
119 REM SELECT GAME SCREEN
120 GRAPHICS N2+N16:GOSUB 4500:SETCOLO
R N,N7,N6:SETCOLOR N1,N3,N8:SETCOLOR N
2,13,N10:SETCOLOR N3,N10,N12:G=N
130 T=N:? #N6;"  SCORE:";SC;:POKE 85,1
3:? #N6;"|HI:|";HS:POSITION N5,N3:? #N
6;"|breakthru|"
140 ? #N6:? #N6;" press option/select"

150 ? #N6;"   |NO. OF BALLS:| 3":B=N3:
L=N7:D=N:POKE 764,62:POSITION N8,N9:?
#N6;"BREAKOUT"
160 GOSUB 4000:C=PEEK(CSL):X=STRIG(N):
IF C=N5 OR C=N6 OR X=N THEN 190
```

```
170 D=D-0.2:D=D*(D>N):SOUND N,F,N10,D:
SOUND N1,F-N1,N10,D:IF D>N9 THEN 160
175 IF PEEK(KB)=58 THEN 3500
180 GOTO 160
190 IF C=N6 OR X=N THEN 230
200 IF D>N9 THEN 170
210 D=N10:B=B+N1:IF B>N5 THEN B=N1
220 POSITION 17,N7:? #N6;B:GOTO 160
230 POKE 53761,N:POKE 53763,N:B=B+N1:G
OSUB 4020:GOSUB 3550
240 POKE 205,N:POKE 203,MSL-INT(MSL/W)
*W:POKE 204,INT(MSL/W):GOSUB 4100
250 POKE 208,N:POKE 209,N6:POKE 21,120
:POKE 22,N1:POKE 24,76:POKE 25,44:POKE
 26,N:POKE 82,N:POKE 83,39
260 POKE 27,N1:POKE 1024,N1:POKE 1025,
N1:POKE 1026,N:POKE 1027,N16
270 POKE 1038,N2:POKE 1039,N2:POKE 104
6,B:POKE 1784,B+N16:POKE 1047,N:POKE 1
048,96
279 REM SETUP DLIST
280 GRAPHICS N4:POKE 559,N:GOSUB 4500:
COLOR N1:FOR A=N TO N3:PLOT N4+A,N:DRA
WTO N4+A,39:PLOT 72+A,N
290 SETCOLOR N,N7,N8:DRAWTO 72+A,39:NE
XT A:POKE DL,112:POKE DL+N1,112:POKE D
L+N2,112:SETCOLOR N1,N11,N8
300 PLOT N4,N:DRAWTO 72,N:POKE DL+N3,7
3:POKE DL+N4,PEEK(88):POKE DL+N5,PEEK(
89):SETCOLOR N3,N3,N8
310 FOR A=DL+N6 TO DL+N11:POKE A,N9:NE
XT A:POKE DL+N12,137:IF G=N3 THEN POKE
 1048,N
320 POKE DL+13,202:POKE DL+14,N:POKE D
L+15,N6:FOR A=DL+N16 TO DL+N16+N5:POKE
 A,138:NEXT A
330 POKE DL+21,201:A=PEEK(88)+PEEK(89)
*W+150:POKE DL+22,A-INT(A/W)*W:POKE DL
+23,INT(A/W)
340 FOR A=DL+24 TO DL+48:POKE A,N9:NEX
```

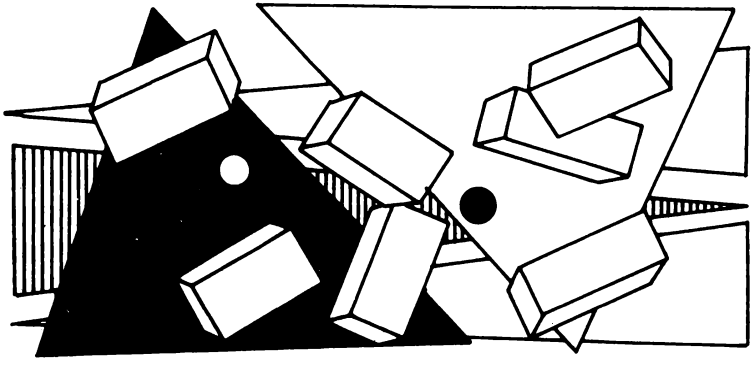
```
T A:POKE DL+49,70:POKE DL+50,234:POKE
DL+51,N6
350 POKE DL+52,70:POKE DL+53,PEEK(660)
:POKE DL+54,PEEK(661):POKE DL+55,N6:PO
KE DL+56,65
360 POKE DL+57,DL-INT(DL/W)*W:POKE DL+
58,INT(DL/W):POKE 560,PEEK(DL+57):POKE
 561,PEEK(DL+58):POKE 559,46
370 FOR A=1536 TO 1636 STEP N20:POKE A
,N:POKE A+19,N:POKE A+N1,85:POKE A+18,
85
380 FOR X=A+N2 TO A+N16+N1:POKE X,F:NE
XT X:NEXT A
400 POKE 54279,PMB:POKE 53277,N3:POKE
704,86:POKE 705,184:POKE 623,N4
405 IF G=N3 THEN POKE 1026,N1
410 POKE PLR+84,F:POKE PLR+85,F:POKE 5
3256,N1:POKE 53248,120:POKE 53260,N
420 POKE 656,N:POKE 658,N:POKE 657,25:
? "hi score ";HS;
425 IF G=N3 THEN POKE 657,24:? "  demo
 mode    "
430 POKE 54286,F:POKE 205,N1:X=USR(INI
T,DLI,VBI,VBD,MAIN):SOUND N,N,N,N:SOUN
D N1,N,N,N
440 POKE 205,N:POKE 656,N:POKE 658,N:P
OKE 657,N6:? "|game over| ";:POKE 657,
24:? " PRESS start   ";
450 IF PEEK(CSL)=N3 THEN POKE 657,N4:?
 "|game aborted|"
460 FOR X=N1 TO N4:A$(X,X)=CHR$(PEEK(1
776+X)+32):NEXT X:SC=VAL(A$(N1,N4)):IF
 SC>HS AND G<>N3 THEN HS=SC
470 IF PEEK(CSL)<>N6 THEN 470
480 FOR X=53248 TO 53255:POKE X,N:NEXT
 X:POKE 53277,N
490 GOTO 110
999 GOTO 999
3499 REM GAME OPTIONS
3500 RESTORE 3510:FOR A=VBD+21 TO VBD+
```

```
25:READ D:POKE A,D:NEXT A:POKE 1029,N:
POKE 1030,N:GOSUB 4020:G=N3
3510 DATA 173,10,210,41,7
3520 POKE VBD+158,21:SOUND N,N,N,N:SOU
ND N1,N,N,N:POKE 1029,N:POKE 1030,N:GO
TO 240
3550 RESTORE 3560:FOR A=VBD+21 TO VBD+
25:READ D:POKE A,D:NEXT A:POKE VBD+158
,60
3560 DATA 165,24,56,229,21,170
3570 RETURN
4000 X=PEEK(KB):POSITION N3,N9:IF X=56
 THEN ? #N6;"IFAST!":G=N1:GOTO 4070
4010 IF X=62 THEN ? #N6;"slow":G=N:GOT
O 4070
4015 GOTO 4070
4020 IF G=N1 THEN 4050
4030 POKE 1029,N2:POKE 1030,254:POKE 1
042,255:POKE 1043,N1
4040 GOTO 4070
4050 POKE 1029,N3:POKE 1030,253:POKE 1
042,254:POKE 1043,N2
4070 IF PEEK(CSL)<>N3 THEN RETURN
4080 T=(T=N):POSITION N8,N9:IF T=N THE
N ? #N6;"BREAKOUT    ":GOTO 4095
4090 IF T=N1 THEN ? #N6;"BREAKTHRU":GO
TO 4095
4095 FOR X=N1 TO 30:NEXT X:RETURN
4100 IF T=N THEN 4150
4110 POKE VBD+66,234:POKE VBD+67,234:P
OKE VBD+112,N:RETURN
4150 POKE VBD+66,208:POKE VBD+67,84:PO
KE VBD+112,N1:RETURN
4500 POKE N16,64:POKE 53774,64:RETURN

4999 REM MAIN ROUTINE
5000 DATA 173,24,4,208,126,133,205,169
,96,141,24,4,234,234,234,169,255,160,1
6,153,1,6,153,21,6
5005 DATA 153,41,6,153,61,6,153,81,6,1
```

```
53,101,6,136,208,235,133,22,200,132,27
,169,16,174,253,6
5010 DATA 232,224,26,176,5,142,253,6,2
08,19,141,253,6,174,252,6,232,224,26,1
76,5,142,252,6,208
5015 DATA 3,141,252,6,162,8,142,3,4,20
2,189,115,4,157,41,4,189,123,4,157,57,
4,202,208,241
5020 DATA 160,0,152,140,8,208,162,3,15
7,25,4,202,208,250,169,1,162,3,157,29,
4,202,208,250,169
5025 DATA 188,141,9,4,133,205,174,23,4
,240,71,169,16,172,244,6,200,192,26,17
6,5,140,244,6,208
5030 DATA 51,141,244,6,172,243,6,200,1
92,26,176,5,140,243,6,208,35,141,243,6
,172,242,6,200,192
5035 DATA 26,176,5,140,242,6,208,19,14
1,242,6,172,241,6,200,192,26,176,5,140
,241,6,208,3,141
5040 DATA 241,6,206,23,4,208,187,173,2
,4,208,54,133,205,160,127,145,203,136,
208,251,206,248,6,206
5045 DATA 22,4,240,59,169,1,133,27,169
,76,133,24,169,48,133,25,169,0,133,26,
169,4,141,14,4
5050 DATA 141,15,4,133,22,141,2,4,172,
132,2,208,251,200,132,205,173,31,208,4
1,4,208,3,133,205
5055 DATA 96,174,132,2,189,12,4,141,0,
4,108,95,4,96,-1
5499 REM IMMEDIATE VBLANK
5500 DATA 216,206,1,4,208,27,173,0,4,1
41,1,4,173,120,2,74,74,41,3,170,173,21
,0,24,125,4
5505 DATA 4,221,8,4,240,2,133,21,165,2
1,133,21,141,0,208,165,205,208,3,76,95
,228,206,16,4,208,32
5510 DATA 173,14,4,141,16,4,165,24,166
,26,24,125,18,4,221,20,4,240,4,133,24,
```

```
208,9,165,25
5515 DATA 73,1,133,26,32,97,4,164,25,1
69,0,145,203,200,145,203,165,24,141,5,
208,206,17,4,208
5520 DATA 20,173,15,4,141,17,4,165,25,
166,27,24,125,18,4,133,25,201,19,144,3
1,164,25,169,255
5525 DATA 145,203,200,145,203,192,92,1
44,5,169,0,141,2,4,206,114,4,208,5,162
,0,142,1,210,76
5530 DATA 95,228,162,1,134,27,202,134,
22,32,97,4,24,144,212,-1
5999 REM TABLES
6000 DATA 1,1,1,16,0,2,254,0,0,180,60,
0,2,1,2,2,2,2,255,1,64,190,3,0,96,0,0,
0,0,0,0,0,0,1,1,1,1,1,1,1,1
6005 DATA 1,1,1,1,1,1,1,2,2,1,1,1,1,1,
1,1,3,3,2,2,1,1,1,1,1,1,1,1,2,2,3,3,2,
22,42,62,82,102
6010 DATA 0,0,10,8,5,3,2,1,140,150,156
,163,134,183,179,167
6015 DATA 0,0,169,199,141,0,210,169,20
1,141,1,210,169,5,141,114,4,96,0,0
6020 DATA 1,1,2,3,3,2,1,1,2,1,1,1,1,1,
1,2,-1
6499 REM DEFFERRED VBLANK
6500 DATA 169,0,133,59,165,205,208,3,7
6,98,228,165,25,201,82,208,47,165,27,2
40,43,165,24,56,229
6505 DATA 21,170,236,3,4,176,32,189,41
,4,141,14,4,141,16,4,189,57,4,141,15,4
,141,17,4
6510 DATA 169,0,133,22,133,27,189,25,4
,133,26,32,97,4,165,22,208,84,165,25,2
01,44,176,78,201
6515 DATA 32,144,74,24,105,224,74,170,
189,73,4,133,208,134,0,165,24,24,105,1
92,74,74,74,168,177
6520 DATA 208,240,49,133,22,169,0,145,
208,165,27,73,1,133,27,173,10,210,41,7
```

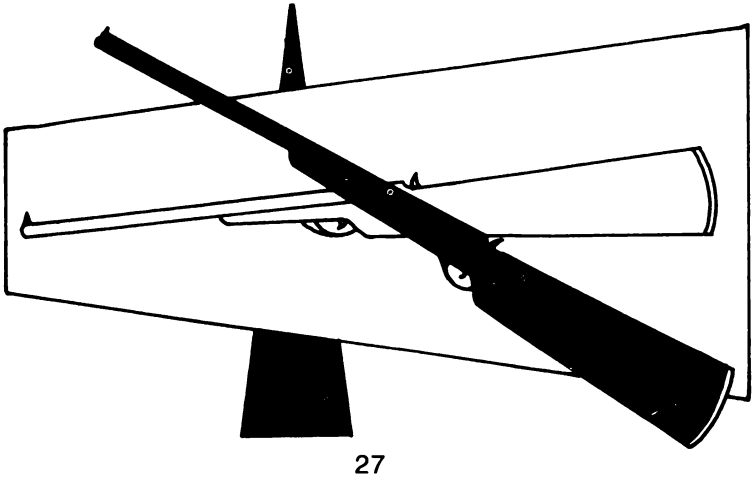
```
,170,189,87,4,141
6525 DATA 0,210,169,167,141,1,210,169,
5,141,114,4,206,24,4,166,0,189,81,4,24
,109,23,4,141
6530 DATA 23,4,165,24,56,233,3,133,60,
76,98,228,-1
6999 REM INITIALIZING ROUTINE
7000 DATA 104,169,0,141,14,212,104,141
,1,2,104,141,0,2,104,141,35,2,104,141,
34,2,104,141,37
7005 DATA 2,104,141,36,2,104,141,96,4,
104,141,95,4,169,192,141,14,212,133,20
5,108,95,4,-1
7499 REM DLIST INTERRUPT
7500 DATA 72,152,72,164,59,192,7,208,4
,160,0,132,59,185,224,6,141,10,212,141
,24,208,200,132,59
7505 DATA 104,168,104,64,-1
7999 REM TABLES
8000 DATA 252,250,248,246,244,242,244,
0,0,0,0,243,227,239,242,229,218,16,16,
16,16,0,98,90,16,0,183,154,16,17
8005 DATA -1
```



SHOOTOUT

(CASSETTA 16K, DISCO 24K)

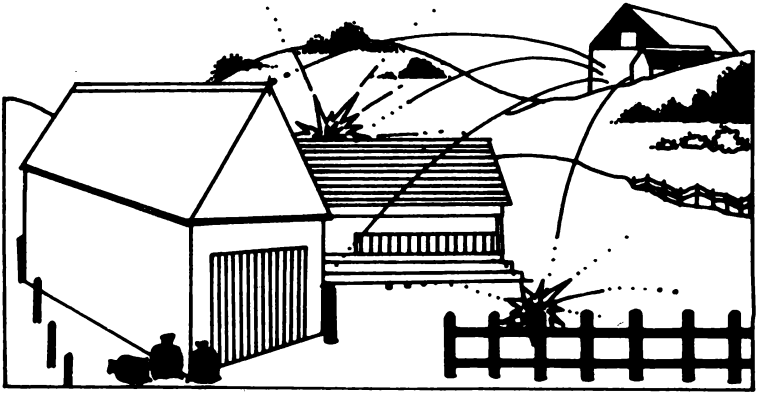
Voi ed il vostro rivale siete confinanti. A causa di una antica faida fra le vostre famiglie, vi sparate a turno l'uno contro l'altro lanciandovi proiettili al di sopra della collina che separa le vostre proprietà.
Dopo aver avviato il programma, sullo schermo apparirà il paesaggio, verrà indicata la velocità del vento e, in basso, troverete scritta la domanda "angle?" ("angolo?").
Il primo giocatore di turno dovrà allora digitare l'angolo che desidera imprimere alla traiettoria di tiro (da zero a 90).
All'apparire della domanda "bags?" ("sacchi?") lo stesso giocatore dovrà indicare il numero di sacchi di polvere da sparo che vorrà utilizzare (di solito da zero a 20). Dopodiché si vedrà la palla di cannone sparata dal giocatore.
Il turno passa poi al secondo giocatore, e cosí via sino a che uno dei due contendenti viene colpito; in quel momento verrà mostrato il punteggio dei due giocatori e un nuovo paesaggio sarà disegnato sullo schermo.



```
5 REM ****  SHOOTOUT  ***
10 REM *** written by ***
15 REM ***  G. Ryan.  ***
20 GOTO 4900
30 T=N1:P=-P:GOSUB 2500
40 SOUND N,N,N,N:Y1=N1:TRAP 40:GOSUB 5
00
50 E=VAL(C$):Y1=N2:GOSUB 500:IF LEN(C$
)>4 THEN IF C$(1,5)="ABORT" THEN 5100
60 IN=VAL(C$)/N2:R=(E-90)*P/57.2958:XI
=IN*SIN(R):YI=IN*COS(R):GOSUB 300
70 X=X((P+N1)/N2):Y=H(X((P+N1)/N2))+N2

80 XT=X:YT=Y
90 IF Y<N THEN 200
100 IF X<N OR X>159 THEN 200
110 SOUND N,F*(F>N),N10,N2:F=F-YI:IF Y
>95 THEN 130
120 COLOR N2:PLOT X,96-Y:COLOR N:PLOT
XT,96-YT:XT=X:YT=Y
130 XI=XI+(WN-XI)*XA:YI=YI-G:X=X+XI:Y=
Y+YI
140 IF X<N OR X>159 THEN 200
150 IF Y>H(X) THEN 90
160 X=X+XI*(H(X)-Y)/YI
170 X=INT(X):IF X>159 THEN X=159
180 Y=H(X):GOTO 200
200 COLOR N:PLOT XT,96-YT:IF INT(X((N1
-P)/N2)/N2)<>INT(X/N2) THEN 30
210 S((P+N1)/N2)=S((P+N1)/N2)+N1:COLOR
 N2
220 FOR Z=-N6 TO N6 STEP N2:PLOT X,95-
Y:DRAWTO X+Z,90-Y:NEXT Z
230 FOR Z=15 TO N STEP -0.6:FOR A=N TO
 2:SOUND A,RND(0)*50+100,8,Z:NEXT A:SE
TCOLOR 4,3,Z-(Z>13)+RND(N)*2:NEXT Z
240 POKE 712,N:FOR A=N TO N2:SOUND A,N
,N,N:NEXT A
250 FOR A=N TO 1000:NEXT A:GOTO 5100
300 FOR A=15 TO N STEP -N2:SOUND N,RND
```

```
(N)*50+120,8,A:NEXT A:F=150
310 RETURN
500 POKE 764,255:POKE 752,N:POKE 657,X
P((P+N1)/N2):POKE 656,Y1:C$=""
510 INPUT C$:POKE 657,XP((P+N1)/N2):PO
KE 752,N1:POKE 656,Y1:? ":"
520 RETURN
2000 POKE 658,N:POKE 657,XP:POKE 656,Y
P
2010 RETURN
2500 COLOR N2:PLOT X(N),96-H(X(N)):DRA
WTO X(N),95-H(X(N))
2510 PLOT X(N1),96-H(X(N1)):DRAWTO X(N
1),95-H(X(N1))
2520 RETURN
4900 RESTORE 4900:DATA 0,1,2,3,6,10
4910 READ N,N1,N2,N3,N6,N10
5000 DIM H(159),X(N1),S(N1),C$(11),CL$
(N1),XP(N1)
5010 X(N)=9:X(N1)=150:S(N)=N:S(1)=N:CL
$=CHR$(125):XP(N)=7:XP(N1)=28:POKE 82,
N2:POKE 83,39
5020 GRAPHICS 23:DL=PEEK(560)+256*PEEK
(561)+N2:H=INT(DL/256):L=DL-H*256:POKE
 559,N:POKE 560,L:POKE 561,H
5030 FOR A=DL+75 TO DL+105:IF PEEK(A)(
)65 THEN NEXT A
5040 POP :A=A-5:POKE A,66:POKE A+N1,PE
EK(660):POKE A+N2,PEEK(661):POKE A+3,N
2:SETCOLOR N2,N,N
5050 POKE A+4,N2:POKE A+5,N2:POKE A+N6
,65:POKE A+7,L:POKE A+8,H
5060 POKE 559,34:POKE 659,N:POKE 703,4
:P=N1
5070 SETCOLOR N,15,N6:SETCOLOR N1,N3,1
2:SETCOLOR N2,15,N
5100 POKE 752,N1:? #N6;CL$:? CL$;:POKE
 657,N3:? S(N);:POKE 657,N10:? "SHOOTO
UT by G. Ryan";:POKE 657,36
5110 ? S(N1):RAD :MH=RND(N)*75+5:LE=IN
```

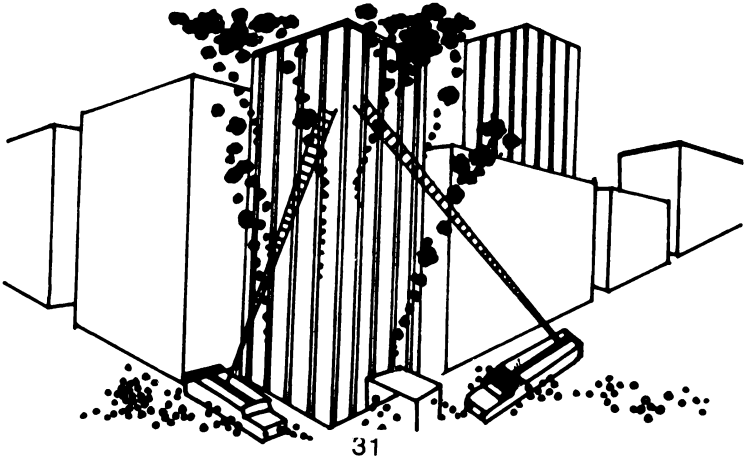
```
T(RND(N)*25)+8:RE=INT(RND(N)*25)+8
5120 MW=10*RND(N)+6:XH=85*RND(N)+38:WN
=16*RND(N)-8:G=0.1:XA=3.0E-03
5130 A=XH/MW:B=1/MW:C=LE/MH:D=RE/MH:CO
LOR N1
5140 FOR I=N TO 159:J=A-I*B:IF J*-J<-2
93 THEN J=17.1172427
5150 Y=MH*EXP(-J*J):H=LE+Y-Y*C:IF I>XH
 THEN H=RE+Y-Y*D
5160 H=H+INT(RND(N)*2.5)
5180 PLOT I,95:DRAWTO I,95-H:H(I)=H:NE
XT I:? CL$;
5190 XP=N2:YP=N1:GOSUB 2000:? "Angle:"
;:POKE 657,23:? "Angle:"
5200 XP=N3:YP=N2:GOSUB 2000:? "Bass:";
:POKE 657,24:? "Bass:"
5210 Z=INT(4*ABS(WN)+0.5):XP=17:YP=N:G
OSUB 2000:IF WN>N THEN ? "----";Z;"-->"
:GOTO 5230
5220 ? "<--";Z;"----"
5230 XP=4:YP=N:GOSUB 2000:? "PLR (1) "
;S(N);
5240 POKE 657,28:? "PLR (2) ";S(N1):PO
KE 752,N:GOTO 30
```



TOWER OF FLAMES

(CASSETTA 16K, DISCO 24K)

Un grattacielo è in fiamme e, in un disperato tentativo di sopravvivere, i suoi occupanti si gettano dalla sommità dell'edificio. Come capo dei pompieri dovete cercare di salvarli prima che muoiano sfracellandosi al suolo. Dovete afferrarli al volo nel vostro telo; ogni persona mancata naturalmente morirà: se ne mancherete tre sarete licenziati.
Usate il POKE per caricare il linguaggio macchina ed il computer farà apparire il titolo. Premete SELECT per scegliere il livello al quale giocare e premete START per iniziare il gioco. Per spostare il telo adoperate il joystick in "port 1".
Per ogni individuo salvato otterrete 10 punti da moltiplicare per il livello al quale giocate. Dopo tre errori il programma ritorna all'inizio mostrando il punteggio totalizzato.



```
10 REM **TOWER OF FLAMES
20 REM Written by C.McConnell,1984
100 MYPMBASE=0:PPOS=206:PHITE=207:DEDF
LAG=208:VFLAG=209
210 TOP=PEEK(740):PMBASE=(TOP-8)*256:C
HBASE=(TOP-12)*256:START=(TOP-10)*256
310 IVBLANK=START:USER=START+530
360 POKE 106,TOP:GRAPHICS 0:POKE 106,T
OP-3:GRAPHICS 0
410 POKE 106,TOP-5:GRAPHICS 18
430 POSITION 2,2:? #6;"TOWER OF FLAMES
"
450 POSITION 4,4:? #6;"please wait"
460 GOTO 810
500 REM ML DATA
510 DATA 165,209,240,3,76,95,228,173,4
8,6,74,74,74,74,141,80,6,173,48,6,41,1
5,24,109,49,6
511 DATA 141,49,6,201,16,144,8,41,15,1
41,49,6,238,80,6,162,3,189,50,6,240,56
,189,62,6,221
512 DATA 58,6,176,17,169,0,157,50,6,16
9,1,157,54,6,169,2,157,70,6,208,71,189
,62,6,56,237
513 DATA 80,6,157,62,6,222,74,6,208,56
,169,4,157,74,6,189,70,6,73,1,157,70,6
,24,144,40
514 DATA 189,54,6,240,53,165,206,56,23
3,8,221,62,6,176,107,24,105,16,221,62,
6,144,99,165,207,221
515 DATA 66,6,176,92,24,105,4,221,66,6
,144,84,176,6,24,144,103,24,144,149,16
9,0,157,54,6,188
516 DATA 66,6,169,15,208,2,240,88,141,
81,6,138,24,101,0,24,105,4,133,205,169
,0,133,204,145,204
517 DATA 136,206,81,6,208,248,173,84,6
,160,8,24,113,88,201,26,144,14,56,176,
2,16,198,233,10,145
518 DATA 88,169,1,136,208,235,145,88,2
```

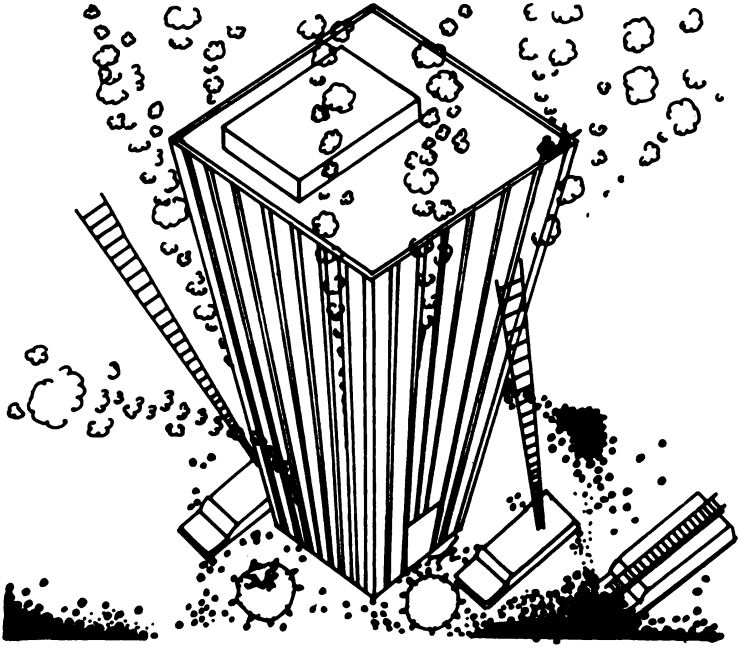
```
06,82,6,208,78,230,209,76,95,228,189,6
6,6,201,220,144,9,169
519 DATA 1,133,209,133,208,76,95,228,2
4,109,80,6,157,66,6,24,144,47,206,85,6
,208,42,173,83,6
520 DATA 141,85,6,169,1,157,50,6,169,0
,157,54,6,173,10,210,41,127,24,105,50,
157,58,6,169,63
521 DATA 157,66,6,169,220,157,62,6,169
,0,157,70,6,202,16,157,173,0,211,106,1
76,4,198,207,198,207
522 DATA 106,176,4,230,207,230,207,166
,207,224,206,144,4,160,206,132,207,224
,158,176,4,160,158,132,207,106
523 DATA 176,20,174,86,6,198,206,202,2
08,251,165,206,201,48,176,27,169,48,13
3,206,208,21,106,176,18,174
524 DATA 86,6,230,206,202,208,251,165,
206,201,210,144,4,169,210,133,206,165,
206,141,4,208,24,105,2,141
525 DATA 5,208,24,105,2,141,6,208,24,1
05,2,141,7,208,165,0,24,105,7,133,205,
169,0,133,204,162
526 DATA 3,189,50,6,24,125,54,6,240,41
,138,72,188,66,6,189,70,6,10,10,10,10,
24,105,15,170
527 DATA 169,16,141,81,6,189,0,6,145,2
04,136,202,206,81,6,208,244,104,170,18
9,62,6,157,0,208,198
528 DATA 205,202,16,201,169,0,164,207,
136,136,145,204,200,145,204,200,169,25
5,145,204,200,145,204,200,169,0
529 DATA 145,204,200,145,204,206,92,6,
208,23,173,93,6,141,92,6,172,10,210,17
7,88,201,130,240,4,201
530 DATA 131,208,4,73,1,145,88,76,95,2
28,169,0,133,209,104,165,209,240,252,9
6,-1
600 REM CHSET DATA
```

```
610 DATA 255,129,129,129,129,129,129,2
55
620 DATA 64,98,243,247,247,255,127,126

630 DATA 0,32,114,119,119,127,127,62
640 DATA -1
700 REM PAGE 6 DATA
710 DATA 0,0,0,24,24,24,24,8,8,56,8,24
,56,40,40,64
720 DATA 0,0,0,24,24,24,24,8,8,24,24,2
4,24,8,8,24
730 DATA 0,0,0,24,153,153,90,66,60,60,
24,60,36,36,36,36
800 REM **READ IN ML
810 TRAP 2030
2010 FOR K=START TO START+9999:READ A:
POKE K,A:NEXT K
2020 REM **READ IN CHARACTER SET
2030 POP :TRAP 3030
2040 FOR T=0 TO 511:POKE CHBASE+T,PEEK
(57344+T):NEXT T
3010 FOR K=CHBASE+8 TO CHBASE+9999:REA
D A:POKE K,A:NEXT K
3020 REM **READ IN PAGE 6 DATA
3030 POKE 623,17:POKE 756,CHBASE/256:P
OP :TRAP 4030
4010 FOR K=1536 TO 1793:READ A:POKE K,
A:NEXT K
4030 POP
4510 LEVEL=1
4710 POKE 54279,TOP-8:POKE 53277,3:POK
E 559,62:POKE MYPMBASE,TOP-8:HS=0:SC=0

5020 POKE 708,6:POKE 709,72:POKE 710,5
2:POKE 711,78
5050 ? #6;CHR$(125)
5060 POSITION 0,0:? #6;"score";SC
5070 POSITION 0,1:? #6;"high score ";H
S
5090 POSITION 13,0:? #6;"level";LEVEL;
" "
```

```
5110 POSITION 0,3:? #6;"press start/se
lect"
5510 K=PEEK(53279):T=PEEK(53264)
5600 IF K=6 OR T=0 THEN 6110
5700 IF K=5 THEN LEVEL=LEVEL*(LEVEL(9)
+1:POSITION 18,0:? #6;LEVEL;" "
5710 IF PEEK(53279)=K THEN 5710
5800 GOTO 5510
6000 REM **START GAME
6110 POKE VFLAG,1:POKE 54286,0
6140 HI=INT(IVBLANK/256):POKE 547,HI:P
OKE 546,IVBLANK-HI*256
6180 POKE 54286,64
7010 STAGE=LEVEL:SC=0:MEN=3:POSITION 5
,0:? #6;"00000"
7610 POSITION 0,1:? #6;"
  "
7710 POSITION 0,3:? #6;"
     "
7740 POSITION 13,0:? #6;"left:";MEN
8000 REM **SET DIFFICULTIES
8110 K=STAGE:IF K>10 THEN K=10
8120 POKE 1620,K:K=100-STAGE*2:IF K<50
 THEN K=50
8130 POKE 1619,K:K=14+STAGE*2
8140 POKE 1584,K:N=K/4:IF N>10 THEN N=
10
8150 POKE 1622,N
8210 POKE 1618,6+STAGE*2
8310 POKE 1628,1:POKE 1629,1
8500 REM **SETUP SCREEN
8510 COLOR 33:FOR T=2 TO 10:PLOT 0,T:D
RAWTO 19,T:NEXT T
8530 COLOR 95:PLOT 0,11:DRAWTO 19,11
8610 COLOR 162:FOR T=0 TO 30:X=RND(0)*
18:Y=RND(0)*8+2:PLOT X,Y:NEXT T
8920 POSITION 6,5:? #6;"STAGE ";STAGE
8940 SOUND 0,100,10,10:FOR T=0 TO 70:N
EXT T:SOUND 0,200,10,10:FOR T=0 TO 70:
NEXT T
```

```
8960 SOUND 0,150,10,10
9000 REM **SET ML VARIABLES
9010 FOR T=0 TO 1:POKE PMBASE+768+PEEK
(PHITE)+T,0:NEXT T
9030 POKE PHITE,200:POKE PPOS,100
9110 FOR T=0 TO 3:FOR K=0 TO 15:POKE P
MBASE+1024+T*256+PEEK(1602+T)-15+K,0:N
EXT K:NEXT T
9210 FOR T=1586 TO 1609:POKE T,0:NEXT
T
9260 FOR T=1610 TO 1613:POKE T,1:NEXT
T
9500 REM **SET BASIC VARIABLES(FUEL..)

9700 COLOR 33:PLOT 6,5:DRAWTO 13,5
9710 REM **ZERO PLAYER
9750 SOUND 0,0,0,0
10000 REM **HERE WE GO
10010 POKE 704,16*INT(16*RND(0))+14
10020 POKE 705,16*INT(16*RND(0))+14
10030 POKE 706,16*INT(16*RND(0))+14
10035 POKE 707,16*INT(16*RND(0))+14
10040 POKE DEDFLAG,0:A=USR(USER):SOUND
 0,0,0,0:SOUND 1,0,0,0
11010 IF PEEK(DEDFLAG) THEN 15010
12000 REM **END OF STAGE
12010 K=0
12110 SC=SC+K:POSITION 5,0:? #6:SC
12210 STAGE=STAGE+1
12310 GOTO 8110
15000 REM **PLAYER DIED
15010 MEN=MEN-1
16000 REM **EXPLOSION
16010 FOR K=0 TO 4:IF PEEK(1602+K)<220
 THEN NEXT K
16020 POP
16030 FOR T=0 TO 15:FOR G=15 TO T STEP
 -1
16035 SOUND 0,T*16+G,10,10
16040 X=PMBASE+1024+256*K+205+G:POKE X
```

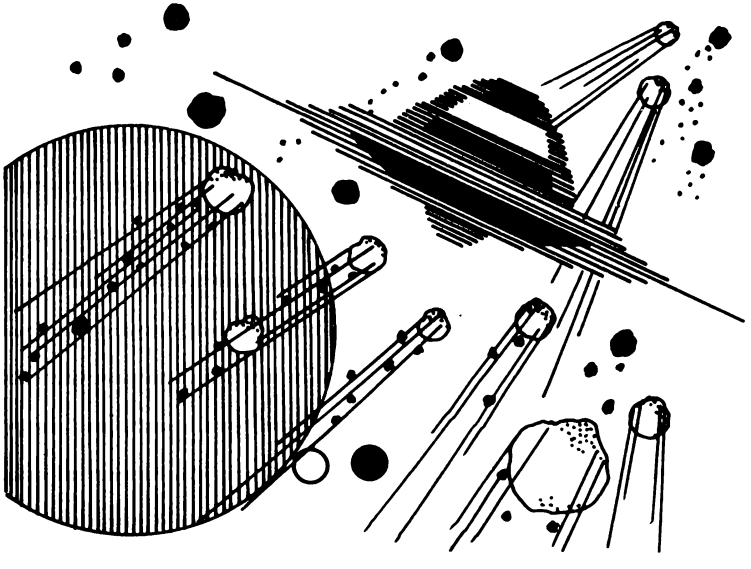
```
,PEEK(X-1):NEXT G:NEXT T
17000 REM **DISPLAY NUMBER OF MEN LEFT

17110 POSITION 13,0:? #6;"LEFT:";MEN
17210 SOUND 0,0,0,0
18000 IF MEN>0 THEN 9010
19000 REM **GAME OVER
19010 POSITION 5,6:? #6;"game over":FO
R K=1 TO 150:NEXT K
19020 COLOR 33:PLOT 5,6:DRAWTO 13,6
19030 SCRN=PEEK(88)+256*PEEK(89)+5:SC=
(PEEK(SCRN)-16)*1000+(PEEK(SCRN+1)-16)
*100+(PEEK(SCRN+2)-16)*10
19040 SC=(SC+(PEEK(SCRN+3)-16))*10
19050 IF SC>HS THEN HS=SC
19060 GOTO 5070
```



ASTEROID STORM

(CASSETTA 16K, DISCO 16K)

Sulla rotta per Alpha Centauri, siete incappati in una tempesta di asteroidi. Dovete farvi strada sino all'estremo opposto senza venire colpiti da questa pioggia di rocce. Avviate il linguaggio macchina con il POKE ed il computer farà apparire il titolo. Premete START per cominciare il gioco e adoperate il joystick in "port 1" per spostarvi a destra o a sinistra; premete il pulsante FIRE per accelerare in avanti.
Se colpite le capsule che si trovano fra le rocce otterrete dei punti extra; altri punti vi verranno assegnati se riuscirete ad attraversare sani e salvi la tempesta. Ogni 10.000 punti vi verrà assegnata un'altra astronave.



```
10 REM ***ASTEROID   STORM***
20 REM **BY CLIFF MCCONNELL**
50 REM **VARIABLES
60 N0=0:N1=1:N2=2:N3=3:N6=6:DIM H(N3),
I(N3):I(N0)=N3:I(N1)=N6+N6:I(N2)=48:I(
N3)=192
65 FOR G=N0 TO N3:H(G)=N0:NEXT G
70 DELAY=14552:WAIT=14553:VFLAG=14554:
DFLAG=14555:AFLAG=14556:BFLAG=14557:DL
I=14376:USER=14399:START=14080
80 M=14336:BASE=203:PPOS=205:PHITE=206
:SDELAY=208:SWAIT=209:HFLAG=14558
100 REM **SETUP GRAPHICS
110 POKE VFLAG,N1:POKE 106,59:GRAPHICS
 18:POKE BASE+N1,M/256+N2+N2:POKE BASE
,N0:POKE 623,16:HS=N0
115 POSITION N3,N3:? #N6;"ASTEROID STO
RM":POSITION N3+N1,N6:? #N6;"PLEASE  W
AIT":POKE 53774,N0
120 DLIST=PEEK(560)+256*PEEK(561):POKE
 DLIST+N3,PEEK(DLIST+N3)+128
130 FOR K=N6 TO 16:POKE DLIST+K,PEEK(D
LIST+K)+128:NEXT K
140 POKE SWAIT,N6:POKE 53277,N3:POKE 5
59,62:POKE 54279,56:POKE PHITE,240:POK
E PPOS,100
145 FOR G=N0 TO 255:POKE M+768+G,N0:PO
KE M+1280+G,N0:POKE M+1536+G,N0:POKE M
+1792+G,N0:NEXT G
150 FOR K=N1 TO 12:RESTORE :FOR L=N1 T
O 14:READ A:POKE M+1297+L+K*16,A:POKE
M+1553+L+K*16,A
160 NEXT L:NEXT K
180 POKE 705,N0:POKE 706,N1:POKE 704,1
16:POKE 707,116
200 REM **ASTEROIDS
210 DATA 0,28,22,50,98,195,129,193,67,
70,100,36,60,0
300 REM **PLAYERS SHIP
310 DATA 8,28,28,62,62,62,127,0,0,0,0,
```

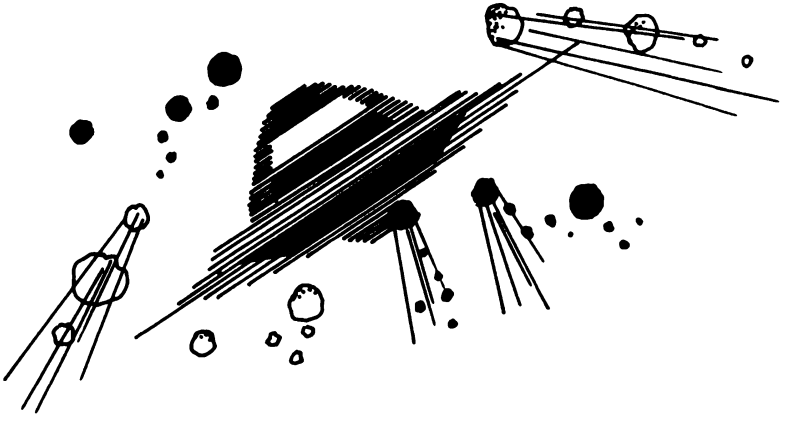
```
0,0,0,0,0
320 FOR K=N0 TO 15:READ A:POKE 14536+K
,A:NEXT K

500 REM **POKE IN MACHINE LANGUAGE
550 N=N0:RESTORE 32000
560 READ A:IF A<>-N1 THEN POKE START+N
,A:N=N+N1:GOTO 560
600 REM **SETUP INTERUPTS
620 A=USR(START):A=INT(DLI/256):POKE 5
13,A:POKE 512,DLI-256*A:POKE 54286,192

630 FOR G=N0 TO 13:POKE 705,G+240:POKE
 706,G:FOR K=N0 TO 15:NEXT K:NEXT G
650 ? #N6;CHR$(125);:POSITION N1,N6:?
#N6;"PLEASE PRESS START":IF SC>HS THEN
 HS=SC
660 POSITION N1,N0:? #N6;"score ";SC:P
OSITION N1,N1:? #N6;"high score ";HS:N
E=1
670 IF PEEK(53279)<>N6 THEN 670
690 ? #N6;CHR$(125):SC=N0:STAGE=N1:POS
ITION N1,N0:? #6;"score ";SC:L=N3:GOSU
B 12010
700 REM **SET ASTEROID SPEEDS
710 POKE WAIT,N2:POKE DELAY,N1:POKE VF
LAG,N1:POSITION N6,N6:? #N6;"STAGE ";S
TAGE;:POKE HFLAG,N0:BS=N1
715 RESTORE 13030:GOSUB 13010
717 GOSUB 10010
720 FOR K=N0 TO 11
730 X1=INT(RND(N0)*STAGE*4)+INT(RND(N0
)+0.5)*128+N1:X2=INT(RND(N0)*STAGE*4)+
INT(RND(N0)+0.5)*128+N1
740 POKE 14464+K,X1:POKE 14500+K,X2:NE
XT K
750 POKE VFLAG,N0:POKE PHITE,240
1000 REM **GAME STARTS
1010 POKE SDELAY,60:POKE AFLAG,N0
1030 POSITION N6,N6:? #N6;"              "
```

```
1040 POKE 53278,N0:POKE DFLAG,N0:POKE
AFLAG,N0:POKE VFLAG,N0
1100 A=USR(USER):ON PEEK(BFLAG) GOTO 1
510,1410,1220
1200 REM **MADE IT!
1220 GOSUB 11010
1250 STAGE=STAGE+N1:GOTO 710
1400 REM **HIT BONUS POD
1410 N=N0
1420 IF PEEK(53256+N)/N2<>INT(PEEK(532
56+N)/N2) THEN 1450
1430 N=N+N1:IF N<N2+N2 THEN 1420
1440 POKE DELAY,N1+N1:GOTO 1040
1450 H=(PEEK(PPOS)-40)/8:I=(PEEK(PHITE
)-52)/16
1455 K=100*INT(2*BS+0.5):POSITION H,I:
? #N6;K:SC=SC+K:BS=BS+N1
1460 FOR K=1200 TO N0 STEP -20:SOUND N
0,K,10,N2+N2:NEXT K
1470 FOR K=N0 TO N3:G=M+768+H(N)+K:POK
E G,PEEK(G)-I(N):NEXT K
1480 POSITION H,I:? #N6;"       ":POSIT
ION 7,N0:? #N6;SC
1490 SOUND N1,N0,N0,N0:GOSUB 3000
1495 GOTO 1440
1500 REM **HIT ASTEROID
1510 L=L-N1:GOSUB 2010:IF L<>N0 THEN 1
520
1515 POKE 53278,N0:POSITION N2+N3,N6:?
 #N6;"GAME OVER":FOR G=N0 TO 150:NEXT
G:GOTO 650
1520 N=PEEK(PHITE)
1530 POKE SDELAY,N0:POKE DELAY,N0:POKE
 PHITE,35+16*L:POKE PPOS,51:POKE HFLAG
,N1:POKE VFLAG,N0:G=PEEK(20)
1550 IF PEEK(20)=G THEN 1550
1560 POKE VFLAG,N1:GOSUB 12010:POKE DF
LAG,N1:POKE HFLAG,N0:POKE VFLAG,N0
1570 POKE SDELAY,N0:POKE DELAY,N0:IF P
EEK(PHITE)<N THEN 1570
```

```
1580 GOSUB 11010
1590 POKE AFLAG,N0:GOTO 700
2000 REM **EXPLOSION
2010 POKE AFLAG,N1:POKE VFLAG,N0:POKE
HFLAG,N1:POKE SDELAY,N0:POKE DELAY,N0
2020 FOR G=15 TO 8 STEP -N1:FOR K=N1 T
O G:SOUND N0,255-((G-8)*32)-K,10,10
2030 POKE 14536+K-N1,PEEK(14536+K):NEX
T K:NEXT G:POKE VFLAG,N1:POKE PHITE,PE
EK(PHITE)-N6-N2
2040 DATA 0,73,73,42,42,28,28,119,119,
28,28,42,42,73,73,0
2050 RESTORE 2040:FOR G=N0 TO 15:READ
A:POKE 14536+G,A:NEXT G:POKE VFLAG,N0:
G=PEEK(20)
2060 IF G=PEEK(20) THEN 2060
2070 POKE VFLAG,N1:FOR G=200 TO N0 STE
P -10:SOUND N0,G,10,10:NEXT G
2080 RESTORE 310:FOR G=N0 TO 15:READ A
:POKE 14536+G,A:NEXT G
2090 RETURN
3000 REM **EXTRA
3010 POKE VFLAG,N1:IF SC<NE*10000 THEN
 3050
3020 NE=NE+1:IF L>10 THEN 3050
3030 FOR G=N1 TO N3+N1:SOUND N0,25,10,
12:FOR K=N1 TO 10:NEXT K:SOUND N0,N0,N
0,N0:FOR K=N1 TO 40:NEXT K:NEXT G
3040 L=L+1:GOSUB 12010
3050 POKE VFLAG,N0:RETURN
10000 REM **PUT BONUS PODS ON SCREEN
10010 FOR G=N0 TO N3:FOR K=N0 TO N3:PO
KE M+768+H(G)+K,N0:NEXT K:NEXT G
10030 FOR G=N0 TO N3:H(G)=INT(RND(0)*1
55)+45:H=INT(RND(0)*151)+45
10040 FOR K=N0 TO N3:N=M+768+H(G)+K:PO
KE N,PEEK(N)+I(G):NEXT K
10050 POKE 53252+G,H:NEXT G
10060 RETURN
11010 POKE DFLAG,N1:POKE AFLAG,N1:POKE
```

```
 VFLAG,N0
11020 IF PEEK(PHITE)>239 THEN 11050
11030 SC=SC+10*STAGE:POSITION 7,N0:? #
N6;SC;:GOSUB 3000
11040 SOUND N0,50,10,10:FOR K=N1 TO N2
+N3:NEXT K:SOUND N0,N0,N0,N0:POKE DELA
Y,N0:GOTO 11020
11050 POKE DFLAG,N0:RETURN
12000 REM **DISPLAY SHIPS
12010 FOR G=N1 TO L-N1:RESTORE 310:FOR
 K=N0 TO 15:READ A:POKE M+1792+20+K+16
*G,A:NEXT K:NEXT G
12020 FOR K=N0 TO 15:POKE M+1792+20+K+
16*L,N0:NEXT K
12030 POKE 53251,51:RETURN
13000 REM **TUNE
13010 READ A,B:IF A<>-N1 THEN SOUND N0
,A,10,10:FOR G=N1 TO B*25:NEXT G:GOTO
13010
13020 SOUND N0,N0,N0,N0:RETURN
13030 DATA 121,2,0,.1,121,.33,108,.33,
96,.33,91,2,-1,-1
32000 DATA 162,55,160,11,169,6,32,92,2
28,104,96,173,218,56,240,3,76,224,55,1
73,220,56,208,63,162,0
32001 DATA 189,128,56,133,62,189,140,5
6,133,4,189,152,56,133,0,32,231,55,165
,4,157,140,56,165,0,157
32002 DATA 152,56,189,164,56,133,62,18
9,176,56,133,4,189,188,56,133,0,32,231
,55,165,4,157,176,56,165
32003 DATA 0,157,188,56,232,224,10,208
,195,173,222,56,208,18,173,0,211,106,1
06,106,176,3,76,105,55,230
32004 DATA 205,106,144,2,198,205,173,2
19,56,240,4,230,206,208,49,173,16,208,
208,26,206,216,56,208,29,173
32005 DATA 217,56,141,216,56,198,206,1
69,132,141,1,210,169,150,141,0,210,76,
158,55,169,0,141,1,210,141
```

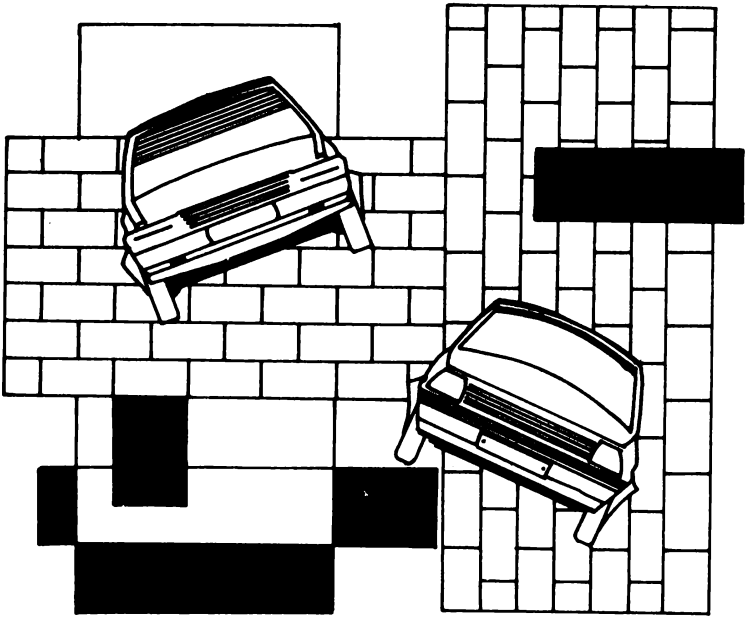
```
32005 DATA 0,210,198,208,208,6,165,209
,133,208,198,206,165,205,201,200,144,4
,169,200,133,205,165,205,201,45
32007 DATA 176,4,169,45,133,205,165,20
6,201,40,176,4,169,40,133,206,160,0,15
2,145,203,136,208,251,164,205
32008 DATA 162,15,189,200,56,145,203,1
36,202,16,247,165,205,141,0,208,169,0,
133,207,76,95,228,165,62,41
32009 DATA 112,74,74,74,74,133,1,165,6
2,41,15,240,15,24,101,4,133,4,201,16,1
44,6,41,15,133,4
32010 DATA 230,1,165,62,41,128,208,14,
165,0,56,229,1,201,35,176,2,169,210,13
3,0,96,165,0,24,101
32011 DATA 1,201,210,144,2,169,35,133,
0,96,72,138,72,166,207,189,152,56,141,
1,208,189,188,56,141,2
32012 DATA 208,230,207,104,170,104,64,
104,173,12,208,240,4,169,1,208,29,173,
8,208,24,109,9,208,109,10
32013 DATA 208,109,11,208,41,1,240,4,1
69,2,208,8,165,206,201,40,208,220,169,
3,238,218,56,141,221,56,96,-1
```



BLOCKOUT

(CASSETTA 16K, DISCO 24K)

Siamo nell'anno 2109 ed i "litigi" fra nazioni vengono risolti con disfide gladiatorie invece che con le guerre. Voi ed il vostro avversario controllate delle "auto della morte" che si lasciano dietro un letale cavo elettrificato. Le auto non possono fermarsi, pertanto lo spazio vitale nell'arena diminuisce sempre di più: dovete cercare di bloccare il vostro avversario togliendogli spazio.
Dopo aver inserito il programma premete SELECT per decidere il numero di partite da effettuare per proclamare il vincitore. Premete OPTION per una partita ad uno o due giocatori e START per iniziare. Adoperate i joystick nelle "port 1" e "port 2" per spostarvi nelle quattro direzioni.



```
0 REM ***  BLOCKOUT     **
2 REM *** written by    **
5 REM *** G. Ryan, 1984 **
10 GOTO 550
20 S=STICK(P):B=B+1:IF S<>15 THEN H(P)
=HI(S):V(P)=VI(S)
30 X=X(P)+H(P):Y=Y(P)+V(P):LOCATE X,Y,
C:COLOR P+1:PLOT X,Y:X(P)=X:Y(P)=Y:P=P
L(P):IF  NOT C THEN 20
40 FOR A=15 TO N STEP -0.7:SOUND N,PEE
K(53770),8,A:NEXT A:IF P=PL(P) THEN 60
50 PL(P)=P:B1=B:GOTO 20
60 B=B+B*PL-B1:B=INT(B/20)*20:S(P)=S(P
)+B:POKE 657,6:? "|bonus|";
70 FOR A=B TO N STEP -20:POKE 656,3:PO
KE 657,12:? A;"  ";:POKE 656,N1:POKE 6
57,4+10*P:? S(P)-A
80 SOUND N,80,10,10:FOR X=N1 TO 10:NEX
T X:SOUND N,N,N,N:FOR X=N1 TO 5:NEXT X
:NEXT A
90 IF S(N)>HS THEN HS=S(N)
100 IF S(N1)>HS THEN HS=S(N1)
110 FOR A=N1 TO 300:NEXT A:GOTO 810
500 POKE 82,N1:GRAPHICS 5:POKE 559,N:D
=PEEK(560)+256*PEEK(561):POKE 659,N1
510 FOR D=D+38 TO D+48:IF PEEK(D)<>66
THEN NEXT D
520 POP :POKE D,70:POKE D+3,6:POKE D+4
,6:POKE D+5,6:POKE 559,34
530 SETCOLOR 0,7,8:SETCOLOR 1,12,10
540 SETCOLOR 2,3,6:SETCOLOR 3,1,8:RETU
RN
550 DIM S(1),X(1),Y(1),H(1),V(1),RN(7)
,PL(1),HI(15),VI(15):N=0:N1=1:W=6:GOSU
B 500
560 S(N)=N:S(N1)=N:RESTORE 570:FOR A=N
 TO 7:READ D:RN(A)=D:NEXT A:CSL=53279
570 RESTORE 580:FOR A=N TO 15:READ X,Y
:HI(A)=X:VI(A)=Y:NEXT A:DATA 5,10,15,2
0,25,50,100,200
```

```
580 DATA 0,0,0,0,0,0,0,0,0,0,1,0,1,0,1
,0,0,0,-1,0,-1,0,-1,0,0,0,0,1,0,-1,0,0
600 ? #6;CHR$(125):COLOR 2:PLOT 0,0:DR
AWTO 79,0:DRAWTO 79,39:DRAWTO 0,39:DRA
WTO 0,0
610 RESTORE 1000:Z=N:COLOR 2
620 FOR A=13 TO 21
630 READ D:IF SGN(D)<>-1 THEN PLOT 14+
D,A:READ D:DRAWTO 15+D,A:GOTO 630
640 IF D=-3 THEN Z=Z+1:IF Z<3 THEN RES
TORE 1010:NEXT A
650 NEXT A:Z=0
660 ? CHR$(125):? "WRITTEN BY G. RYAN"
670 ? "|PLEASE PRESS START|":? "    |hi
 score:|";HS
680 POKE 709,Z:Z=Z*(Z<254)+2:SOUND 0,2
56-Z/3,10,3:SOUND 1,255-Z/3,10,3:IF PE
EK(CSL)<>6 AND STRIG(N) THEN 680
690 SOUND N,N,N,N:SOUND N1,N,N,N:IF ST
RIG(N)=N OR PEEK(CSL)=6 THEN 690
700 GRAPHICS 18:POSITION 6,2:? #W;"|bl
ockout|":POSITION 1,5
710 ? #W;"option : 1 PLAYER":POSITION
1,7:? #W;"select : 5 ROUNDS":FOR A=1 T
O 40:NEXT A:D=N
720 D=D-N1:A=PEEK(CSL):D=D*(A<>7):IF D
<1 THEN GOSUB 740:D=20
730 GOTO 720
740 IF A=6 OR STRIG(N)=N THEN RN=RN(RN
)+1:GOSUB 500:GOTO 810
750 IF A=5 THEN 800
760 IF A<>3 THEN 720
770 PL=PL=0:POSITION 10,5:IF PL=1 THEN
 ? #W;"2 PLAYERS":RETURN
780 ? #W;"1 PLAYER ":RETURN
800 RN=RN+1:RN=RN*(RN<8):POSITION 10,7
:? #W;RN(RN);" ROUNDS  ":RETURN
810 PL(N)=1:PL(1)=N:IF PL=N THEN PL(N)
=N
```

```
850 POP :? #W;CHR$(125):COLOR 3:PLOT 0
,0:DRAWTO 79,0:DRAWTO 79,39:DRAWTO 0,3
9:DRAWTO 0,0
860 RN=RN-1:? CHR$(125):? "P1:";S(0);:
POKE 657,11:? "p";CHR$(18);CHR$(26);S(
1):P=0:H=0:B1=0:B=0
870 ? "|ST:|";RN;:POKE 657,11:? "|HI:|
";HS
880 X(0)=9:X(1)=70:Y(0)=19:Y(1)=19:H(0
)=1:H(1)=-1:V(0)=0:V(1)=0
900 IF  NOT RN THEN 930
910 IF STRIG(0) THEN 910
920 GOTO 20
930 ? "|GAME OVER-hit start|"
940 IF PEEK(53279)<>6 THEN 940
950 GOTO 560
1000 DATA 0,5,-1
1010 DATA 0,0,5,5,9,9,26,26,48,48,-3,0
,6,9,9,12,15,20,23,26,26,29,29,32,35,3
9,39,43,43,46,50,-1
1020 DATA 0,0,6,6,9,9,12,12,16,16,19,1
9,26,26,28,28,32,32,36,36,39,39,43,43,
48,48,-1
1030 DATA 0,0,6,6,9,9,12,12,16,16,19,1
9,26,27,32,32,36,36,39,39,43,43,48,48,
-1
1040 DATA 0,0,6,6,9,9,12,12,16,16,19,1
9,26,26,28,28,32,32,36,36,39,39,43,43,
48,48,-1
1050 DATA 0,6,9,9,12,16,19,23,26,26,29
,29,32,36,39,42,48,48,-1
```

SPACE ATTACK

(CASSETTA 16K, DISCO 24K)

Siete il comandante di una stazione spaziale minacciata da una pioggia di asteroidi e da nemici alieni che giungono verso la stazione dalle quattro direzioni: distruggeteli prima che collidano con voi.
Sparate nella direzione desiderata adoperando un joystick in "port 1". Se avete ottenuto un eccellente punteggio, scriverete il vostro nome nel "Libro della Gloria" (Hall of Fame).

```
5 REM *** SPACE ATTACK
6 REM *** by G. Ryan
10 GOTO 140
19 REM MOVE ALIENS
20 COLOR 32:PLOT A1,12:PLOT A2,12:A1=A
1-D1:A2=A2+D2:COLOR C1:PLOT A1,12:COLO
R C2:PLOT A2,12:COLOR 32
30 PLOT 20,A3:PLOT 20,A4:A3=A3+D3:A4=A
4-D4:COLOR C3:PLOT 20,A3:COLOR C4:PLOT
 20,A4
40 IF PEEK(53770)<DF THEN D1=SP1:D2=SP
2:D3=SP3:D4=SP4:C1=CH:C2=C1:C3=C1:C4=C
1
50 GOTO USR(1629,KL AND A1>23 AND A2<1
6.5 AND A3<10.5 AND A4>13)
59 REM SHOOT LASERS
60 SOUND 0,35,10,8:COLOR 18:PLOT 24,12
:DRAWTO A1,12:C1=32:KL=KL-SGN(D1):D1=0
:COLOR 32:PLOT 24,12:DRAWTO A1,12
70 SOUND 0,0,0,0:A1=S1:GOTO 20
80 SOUND 0,35,10,8:COLOR 18:PLOT 16,12
```

```
:DRAWTO A2,12:C2=32:KL=KL-SGN(D2):D2=0
:COLOR 32:PLOT 16,12:DRAWTO A2,12
90 SOUND 0,0,0,0:A2=S2:GOTO 20
100 SOUND 0,35,10,8:COLOR 124:PLOT 20,
10:DRAWTO 20,A3:C3=32:KL=KL-SGN(D3):D3
=0:COLOR 32:PLOT 20,10:DRAWTO 20,A3
110 SOUND 0,0,0,0:A3=S3:GOTO 20
120 SOUND 0,35,10,8:COLOR 124:PLOT 20,
14:DRAWTO 20,A4:C4=32:KL=KL-SGN(D4):D4
=0:COLOR 32:PLOT 20,14:DRAWTO 20,A4
130 SOUND 0,0,0,0:A4=S4:GOTO 20
139 REM INITIALIZE ROUTINE
140 POKE 82,0:POKE 83,39:GOSUB 800:AD=
1570:RESTORE 1000:TRAP 160
150 FOR AD=1570 TO 1777:READ D:POKE AD
,D:NEXT AD
159 REM DOWNLOAD CHARACTER-SET
160 POP :CHSET=(PEEK(106)-12)*256:D=US
R(1570,57344,58367,CHSET):RESTORE 2000
:TRAP 190
169 REM POKE IN NEW CHARACTERS
170 READ D:FOR AD=CHSET+D TO CHSET+D+7
:READ D:POKE AD,D:NEXT AD:GOTO 170
190 DIM NAME$(120),HS(12),CH(4),BL(4),
SP(5),ST(5),BK(4)
200 FOR A=0 TO 12:HS(A)=0:NEXT A:ST(0)
=0:ST(1)=1:ST(2)=1:ST(3)=2:ST(4)=3:ST(
5)=4
210 CH(0)=0:CH(1)=64:CH(2)=96:CH(3)=64
:CH(4)=64:BL(0)=32:BL(1)=64:BL(2)=112:
BL(3)=96:BL(4)=0
220 NAME$(1)=" ":NAME$(120)=" ":NAME$(
2)=NAME$
230 SP(0)=0.5:SP(1)=0.5:SP(2)=0.6:SP(3
)=0.6:SP(4)=1.2:SP(5)=1.5:BK(0)=161:BK
(1)=34:BK(2)=163:BK(3)=154:GOTO 900
239 REM HI SCORE SCREEN
240 GRAPHICS 17:? #6;"    space attack
":? #6;"    !TOP TEN SCORES!"
250 FOR A=0 TO 9:? #6:? #6;" 00000";:P
```

```
OKE 85,PEEK(85)-LEN(STR$(HS(A))):? #6;
HS(A);" ";NAME$(A*12+1,A*12+12):NEXT A

260 RETURN
279 REM START OF GAME
280 WV=0:SC=0:LV=3:EN=50:DF=40:SP=0:ST
=0:N=0:EX=1000:GRAPHICS 0:POKE 559,0
289 REM SET UP NEW DISPLAY LIST, IN HI
-RES. MULTICOLOR TEXT MODE
290 POKE 1536,112:POKE 1537,112:POKE 1
538,112:POKE 1539,66:POKE 1540,PEEK(88
)
300 POKE 1541,PEEK(89):FOR AD=1542 TO
1564:POKE AD,4:NEXT AD:POKE AD,65:POKE
 1566,0:POKE 1567,6
310 GRAPHICS 0:POKE 752,1:SETCOLOR 2,0
,0:POSITION 11,11:? "{PREPARE FOR ATTA
CK!}"
320 FOR A=1 TO 3:SOUND 0,100,10,8:POKE
 755,2:SETCOLOR 2,3,2:FOR D=1 TO 90:NE
XT D:N=1
330 SOUND 0,160,10,8:POKE 755,0:SETCOL
OR 2,0,0:FOR D=1 TO 90:NEXT D:NEXT A:S
OUND 0,0,0,0:POKE 755,2
339 REM INCREASE DIFFICULTY
340 WV=WV+1:POKE 559,0:POKE 1536,112:G
OSUB 960:POKE 560,0:POKE 561,6:POKE 75
6,CHSET/256
350 POKE 77,0:IF PEEK(53279)=3 THEN 85
0
360 POKE 559,34:SETCOLOR 0,8,6:SETCOLO
R 1,3,10:SETCOLOR 2,14,2:SETCOLOR 3,11
,10
370 FOR A=0 TO 18:COLOR BK(RND(0)*3):P
LOT A,RND(0)*22+1:PLOT A+20,RND(0)*22+
1:NEXT A:IF N THEN 420
380 DF=DF+10:IF DF>160 THEN DF=70
390 IF WV>2 THEN SP=SP+1:IF SP>5 THEN
SP=3
```

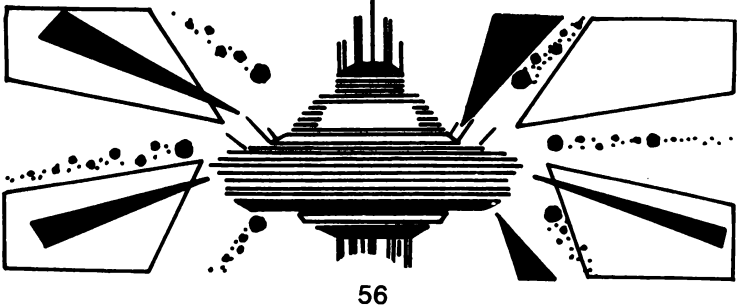
```
400 IF WV>2 THEN ST=ST+1:IF ST>5 THEN
ST=2
409 REM SET ENEMY ATTACK SPEEDS, START
ING POSITIONS, ETC...
410 EN=EN+INT(RND(0)*10+10):IF EN>200
THEN EN=90
420 S1=38-ST(RND(0)*ST):S2=1+ST(RND(0)
*ST):S3=1+ST(RND(0)*ST):S4=23-ST(RND(0
)*ST)
430 N=0:C1=32:C2=C1:C3=C1:C4=C1:A1=38:
A2=0:A3=1:A4=23:KL=EN
440 SP1=SP(RND(0)*SP):SP2=SP(RND(0)*SP
):SP3=SP(RND(0)*SP):SP4=SP(RND(0)*SP)
450 CH=CH(RND(0)*4):D1=0:D2=D1:D3=D1:D
4=D1
459 REM PLAYER'S SHIP
460 POSITION 18,11:? "!jklmn!":POSITIO
N 17,12:? "!opqrstu!":POSITION 18,13:?
 "!vwxyz!"
470 POKE 207,1:POKE 208,1:POKE 209,1:G
OTO 20
480 POSITION 18,11:IF KL=0 THEN 550
489 REM EXPLOSION
490 ? "←TPTPP←":POSITION 17,12:? "←PTP
TPTT←":POSITION 18,13:? "←PTPTT←":D=RN
D(0)*14:IF SC>EX THEN GOSUB 990
499 REM ←P← IS (CTRL-P), ←T← IS (CTRL-
T)
500 FOR A=15 TO 0 STEP -0.3:SOUND 0,15
0,8,A:SOUND 1,255,2,A:SOUND 2,180,0,A:
POKE 1536,BL(RND(0)*4)
510 SETCOLOR 4,D,A+RND(0)*3:NEXT A:SC=
SC+EN-KL:LV=LV-1:POKE 712,0:IF LV<1 TH
EN 600
520 FOR A=1 TO 100:IF PEEK(53279)<>3 T
HEN NEXT A:WV=WV-1:GOTO 310
530 GOTO 950
549 REM ATTACK WAVE OVER
550 GRAPHICS 0:POKE 752,1:SETCOLOR 2,1
```

```
2,2:POSITION 8,11:? "Attack Wave ":WV;
" Completed"
560 POKE 53768,4:POKE 53761,168:POKE 5
3765,176:POKE 53760,254:POKE 53764,127
:SC=INT(SC+EN+EN*SP+INT(RND(0)*100))
570 IF SC<EX THEN FOR A=0 TO 550:NEXT
A:SOUND 0,0,0,0:SOUND 2,0,0,0:GOTO 340

580 GOSUB 980:FOR A=1 TO 550:NEXT A:SO
UND 0,0,0,0:SOUND 2,0,0,0:GOTO 340
600 FOR A=1 TO 300:NEXT A:GRAPHICS 0:S
ETCOLOR 2,0,0:POKE 752,1:GOSUB 960:POS
ITION 15,7
609 REM GAME OVER
610 ? "GAME OVER":POSITION 11,10:? "Yo
ur Score : 00000";
620 POKE 85,PEEK(85)-LEN(STR$(SC)):? S
C:POSITION 7,12
630 ? "The High Score : 00000";:A=SC:I
F A<HS(0) THEN A=HS(0)
640 POKE 85,PEEK(85)-LEN(STR$(A)):? A:
FOR X=1 TO 600:NEXT X
650 IF SC=0 THEN 900
660 FOR A=0 TO 9:IF SC<HS(A) THEN NEXT
 A:GOTO 900
670 FOR D=9 TO A+1 STEP -1:HS(D)=HS(D-
1):NAME$(D*12+1,D*12+12)=NAME$((D-1)*1
2+1,(D-1)*12+12):NEXT D:HS(A)=SC
679 REM ENTER PLAYER'S NAME
680 GRAPHICS 0:SETCOLOR 2,14,0:POKE 75
2,1:GOSUB 960:POSITION 14,6:? "GREAT S
CORE!"
690 POSITION 10,8:? "Now enter your na
me":POSITION 7,10:? "For the high scor
e board"
700 POSITION 12,12:? "←QRRRRRRRRRRRRRR
Q←":CLOSE #5:OPEN #5,4,0,"K":GOSUB 950

709 REM ←Q← IS <CTRL-Q>, ←R← IS <CTRL-
R>
```

```
710 POSITION 12,13:? "|                    |
":POSITION 12,14:? "↑ZRRRRRRRRRRRRRC←
"
719 REM ↑Z← IS (CTRL-Z), ↑C← IS (CTRL-
C)
720 TRAP 720:Z=14:POSITION 13,13:POKE
752,0:? " ";:POKE 764,255
730 POKE 694,0:POKE 702,64:GET #5,X:IF
 X>30 AND X<96 AND Z<26 THEN POKE 767,
0:? CHR$(X);:Z=Z+1:GOTO 730
740 IF (X=126 OR X=30) AND Z>14 THEN ?
 CHR$(X);:Z=Z-1:GOTO 730
750 IF X=125 OR X=156 THEN POKE 752,1:
GOTO 710
760 IF X=127 THEN Z=25:POSITION 24,13:
? CHR$(31);:GOTO 730
770 IF X<>155 THEN 730
780 POKE 752,1:? :FOR Z=1 TO 12:LOCATE
 13+Z,13,X:NAME$(A*12+Z,A*12+Z)=CHR$(X
):NEXT Z
790 CLOSE #5:TRAP 40000:GOTO 900
800 GRAPHICS 18:POSITION 4,4:? #6;"spa
ce attack"
810 POSITION 1,6:? #6;"WRITTEN BY G. R
YAN"
820 RETURN
850 GRAPHICS 18:POSITION 2,4:? #6;"mis
sion aborted"
860 POSITION 5,6:? #6;"|game over|"
870 FOR A=1 TO 250:NEXT A
900 FOR A=53760 TO 53767:POKE A,0:NEXT
 A
910 GOSUB 240:FOR A=1 TO 500:IF STRIG(
0) AND PEEK(53279)<>6 THEN NEXT A:GOTO
 930
920 GOTO 280
930 GOSUB 800:FOR A=1 TO 150:IF STRIG(
0) AND PEEK(53279)<>6 THEN NEXT A:GOTO
 910
940 GOTO 280
```

```
950 POKE 53768,5:POKE 53761,168:POKE 5
3765,176:POKE 53760,254:POKE 53764,127
:RETURN
960 ? CHR$(125);"    SCORE 00000    WAVE
        LIVES";:POKE 85,14-LEN(STR$(SC)
):? SC
970 POSITION 22,0:? WV:POSITION 34,0:?
 LV:RETURN
980 POSITION 15,15:? "EXTRA SHIP!"
990 LV=LV+1:EX=EX+5000:RETURN
999 REM DATA ( MACHINE CODE )
1000 DATA 104,104,133,204,104,133,203,
104,133,206,104,133,205,104,133,208,10
4,133,207,160,0,177,203,145,207
1010 DATA 230,203,208,2,230,204,165,20
4,197,206,208,6,165,203,197,205,240,10
,230,207,208,230,230,208,160
1020 DATA 0,240,224,177,203,200,145,20
7,96
1030 DATA 104,104,133,213,173,197,2,17
4,199,2,141,199,2,142,197,2,104,240,43
,174,120,2,189,161,6
1040 DATA 197,209,240,16,133,209,164,2
08,132,207,133,212,173,31,208,41,252,2
40,9,96,198,207,16,242,169
1050 DATA 20,208,238,169,3,133,213,169
,82,133,212,96,230,213,169,224,208,223
,20,20,20,20,20,60,60
1060 DATA 60,20,80,80,80,20,120,100,20
,-1
1499 REM
1999 REM CHARSET DATA
2000 DATA 256,65,195,20,170,20,195,65,
0
2010 DATA 512,65,20,56,20,150,85,20,65

2020 DATA 640,25,68,17,134,36,226,68,5
0
2030 DATA 672,48,66,45,146,41,82,198,3
5
```

```
2040 DATA 768,52,141,250,39,104,158,19
3,24
2050 DATA 848,0,0,0,0,0,1,42,255
2060 DATA 856,0,0,0,5,85,85,170,255
2070 DATA 864,56,20,85,85,85,85,170,25
5
2080 DATA 872,0,0,0,80,94,85,170,255
2090 DATA 880,0,0,0,0,0,64,168,255
2100 DATA 888,1,5,85,181,181,85,5,1
2110 DATA 896,85,85,85,118,118,85,85,8
5
2120 DATA 904,85,85,85,159,159,85,85,8
5
2130 DATA 912,85,85,85,125,125,85,85,8
5
2140 DATA 920,85,85,85,167,167,85,85,8
5
2150 DATA 928,85,85,85,217,217,85,85,8
5
2160 DATA 936,64,80,85,94,94,85,80,64
2170 DATA 944,255,42,1,0,0,0,0,0
2180 DATA 952,255,170,85,85,5,0,0,0
2190 DATA 960,255,170,85,85,85,85,20,4
4
2200 DATA 968,255,170,85,85,80,0,0,0
2210 DATA 976,255,168,80,0,0,0,0,0
2220 DATA 8,0,0,0,0,0,0,0,192
2230 DATA 16,0,0,0,0,0,4,0,0
2240 DATA 24,0,0,0,64,0,0,0,0
2250 DATA 32,0,192,0,0,0,0,0,0
2260 DATA 992,12,12,12,12,12,12,12,12
```



GOBBLER

(CASSETTA 16K, DISCO 24K)

Siete un tacchino e vi trovate imprigionato in una villa oscura e minacciosa quando vedete un'allucinante massa collosa muovere verso di voi. Questa colla proviene da quattro fantasmi colorati che vogliono liberarsi della vostra presenza. Avete una sola speranza: inghiottire tutte le pillole tonde prima di essere catturato dagli inseguitori. Se inghiottite una "pillola ricostituente" diventate temporaneamente invulnerabili e potrete divorare gli inseguitori per guadagnare punti. Otterrete per ogni pillola inghiottita dieci punti, moltiplicati per il livello di gioco raggiunto.
Azionando il POKE, o venendo catturati per tre volte dai fantasmi, il programma passa ad uno schermo speciale, mostrando il punteggio. Premete SELECT per cambiare il livello di gioco iniziale e START per "divenire tacchino". Adoperate il joystick in "port 1" per spostarvi in alto, in basso, a destra e a sinistra.

```
10 REM ********GOBBLER*********
20 REM **By Cliff McConnell
30 REM **ALL REMS MAY BE OMMITTED
40 VFLAG=1686:CDELAY=1687:CHANGE=1688:
CHANGE2=1689:SINC=1692:HITCLR=53278:CO
NSOL=53279:SPDUPFLAG=1694
50 HITEG0=1677:HPOSG0=1665:PPOS=1664:P
HITE=1676:PSPEED=1669:GSPEED=1670:DEDF
LAG=1682:DOTFLAG=1683:CHAR=1684
110 A=PEEK(740):POKE 106,A-5:START=(A-
12)*256:VBLANK=START+57:MYPMBASE=START
+1024:HS=0:DIM X(4),Y(4)
115 FOR K=1 TO 4:READ X,Y:X(K)=X:Y(K)=
Y:NEXT K
117 DATA 1,1,0,0,1,0,0,1
```

```
120 GRAPHICS 18:POSITION 7,4:? #6;"GOB
BLER":POSITION 1,6:? #6;"by cliff mcco
nnell"
130 GOSUB 30210
140 COLOR 170:PLOT 0,3:DRAWTO 19,3:DRA
WTO 19,7:DRAWTO 0,7:DRAWTO 0,3
150 GOSUB 30010
160 POKE 710,0
170 GRAPHICS 0:DL=PEEK(560)+256*PEEK(5
61):POKE DL+3,70:POKE DL+6,6:POKE DL+7
,6:POKE DL+8,6
210 POKE 704,138:POKE 705,138:POKE 706
,138:POKE 707,138:POKE 711,46:POKE 710
,0:POKE 752,1
220 POKE 559,62:POKE 1681,MYPMBASE/256
:POKE 54279,MYPMBASE/256:POKE 53277,3
310 POKE CDELAY,4:POKE CHAR,0:POKE SIN
C,2
320 LEVEL=1
510 IF SC>HS THEN HS=SC
515 POSITION 0,0:? #6;"score";SC;"
  ":POSITION 11,0:? #6;"high";HS;"
 "
520 POSITION 20,0:? #6;"PRESS START/SE
LECT"
525 STAGE=LEVEL:POSITION 0,1:? #6;"STA
GE ";STAGE:IF PEEK(CONSOL)=5 THEN POSI
TION 32,0:? #6;"select"
527 IF PEEK(CONSOL)=5 THEN 527
528 POSITION 32,0:? #6;"SELECT"
530 IF PEEK(CONSOL)=7 THEN 530
540 IF PEEK(CONSOL)=5 THEN LEVEL=LEVEL
*(LEVEL<9)+1:GOTO 525
550 IF PEEK(CONSOL)<>6 THEN 530
610 POSITION 26,0:? #6;"start"
620 IF PEEK(CONSOL)=6 THEN 620
625 SC=0:GOBS=3
626 ? CHR$(125):PILLS=4+STAGE*2:COLOR
20:POSITION 26,0:? "STAGE ";STAGE:PPIL
LS=1+INT(STAGE/10)
```

```
627 FOR K=1 TO PILLS+PPILLS:SOUND 0,25
5-(K*255/(PPILLS+PILLS)),10,10
628 X=INT(RND(0)*39):Y=INT(RND(0)*19)+
2:LOCATE X,Y,A:IF A<>32 THEN 628
629 IF K>PILLS THEN COLOR 0
635 PLOT X,Y:NEXT K:SOUND 0,0,0,0
637 POKE 704,46:POKE 705,78:POKE 706,1
42:POKE 707,206:POKE 1692,2
640 FOR K=0 TO 3:POKE HITEG0+K,50:POKE
 HPOSG0+K,50+K*50:NEXT K:POKE 1703,0
645 POSITION 0,0:? #6;"score";SC:POSIT
ION 12,0:? #6;"stage";STAGE
660 G=(4+STAGE):K=(2+STAGE*1.2):A=130-
STAGE*3:Q=INT(STAGE/4)
670 IF G>47 THEN G=47
680 IF K>47 THEN K=47
690 IF A<6 THEN A=STAGE
700 IF Q>10 THEN Q=INT(Q/2):GOTO 700
710 POKE CHANGE,A:POKE CHANGE2,Q
720 POKE PSPEED,0:POKE GSPEED,0:POKE P
HITE,200:POKE PPOS,120:POKE VFLAG,1
730 POKE 711,46:A=USR(START):? #6;"
    ":FOR Q=0 TO 200:NEXT Q:POKE PSPEE
D,G:POKE GSPEED,K
740 POKE SPDUPFLAG,6
750 POKE HITCLR,0:POKE DEDFLAG,0:POKE
DOTFLAG,0:POSITION 26,0:? "          ":
POKE 623,0
760 POKE 19,0:POKE VFLAG,0:A=USR(START
+492):POKE VFLAG,1:GOTO 1000+1000*PEEK
(1693)
1000 REM **PLAYER DIED
1010 GOBS=GOBS-1:GOSUB 1510
1020 IF GOBS=0 THEN POSITION 25,0:? "G
AME OVER":FOR G=0 TO 300:NEXT G:GOTO 5
10
1030 POSITION 22,0:? "CHOMPERS LEFT:";
GOBS:FOR G=0 TO 300:NEXT G:POSITION 22
,0:? "                    "
1040 GOTO 640
```

```
1510 POKE 623,16:FOR T=15 TO 0 STEP -0
.2:POKE 711,T:SOUND 0,255-T*17,10,10:N
EXT T
1520 FOR T=250 TO 0 STEP -5:SOUND 0,T,
10,10:NEXT T:SOUND 0,0,0,0:RETURN
2000 REM **PLAYER ATE DOT
2010 POKE 623,0:X=INT((PEEK(PPOS)-46)/
4):Y=INT((PEEK(PHITE)-46)/8)
2011 FOR T=1 TO 4:SOUND 0,T*55,10,10
2013 LOCATE ABS(X+X(T)),ABS(Y+Y(T)),A:
IF A=32 THEN NEXT T:GOTO 750
2014 SOUND 0,0,0,0
2015 IF A=0 THEN 2110
2020 IF A=20 THEN COLOR 32:PLOT X+X(T)
,Y+Y(T):PILLS=PILLS-1:IF PILLS=0 THEN
STAGE=STAGE+1:SC=SC+STAGE*10:GOTO 626
2030 SC=SC+STAGE*10:POSITION 0,0:? "sc
ore";SC:GOTO 750
2110 POKE 1703,1:N=1000/STAGE:HI=INT(N
/256):POKE 1702,HI:POKE 1701,N-HI*256:
POKE 1704,1
2120 COLOR 32:PLOT X+X(T),Y+Y(T):BONUS
=200
2130 POKE 704,120:POKE 705,120:POKE 70
6,120:POKE 707,120:POKE GSPEED,PEEK(PS
PEED)-2:POKE 1692,10
2140 GOTO 750
3000 REM **PILL RAN OUT
3010 POKE 704,46:POKE 705,78:POKE 706,
142:POKE 707,206
3020 POKE 1703,0:POKE 1692,2:GOTO 750
4000 REM **ATE GHOST
4010 N=0:LOC=53257:IF PEEK(53257)=0 TH
EN LOC=53258
4020 IF PEEK(LOC)>7 THEN N=3:GOTO 4100

4030 IF PEEK(LOC)>3 THEN N=2:GOTO 4100

4040 IF PEEK(LOC)>1 THEN N=1
4100 SC=SC+BONUS
```

```
4107 POSITION 28,0:? BONUS
4110 FOR T=150 TO 0 STEP -2:SOUND 0,T,
10,10:NEXT T:POSITION 28,0:? #6;"
 ";:BONUS=BONUS*2
4120 POSITION 0,0:? #6;"score";SC:POKE
 HPOSG0+N,60:IF PEEK(PPOS)<120 THEN PO
KE HPOSG0+N,185
4125 POKE 53248+N,PEEK(HPOSG0+N):POKE
HITCLR,0
4130 GOTO 750
20000 REM *DATA FOR GHOSTS AND PLAYER
20010 DATA 0,0,0,56,124,124,124,254,25
4,254,254,170,170,0,0,0,0,0,0,56,124,1
24,84,214,254,254
20011 DATA 254,84,84,0,0,0,0,0,0,56,12
4,124,84,214,254,254,254,170,170,0,0,0
,0,0,0,56
20012 DATA 124,124,84,214,254,254,254,
84,84,0,0,0,0,0,0,24,60,126,255,255,25
5,255,126,60,24,0
20013 DATA 0,0,0,0,0,24,60,102,231,195
,195,231,102,60,24,0,0,0,0,0,0,24,60,1
02,195,129
20014 DATA 129,195,102,60,24,0,0,0,0,0
,0,24,60,102,231,195,195,231,102,60,24
,0,0,0,-1
30000 REM *POKE IN MACHINE LANGUAGE
30010 RESTORE 32000:N=0
30020 READ A:IF A<>-1 THEN POKE START+
N,A:N=N+1:GOTO 30020
30030 HI=INT(VBLANK/256):POKE START+49
,HI:POKE START+51,VBLANK-HI*256
30040 RETURN
30200 REM *POKE IN CHARACTER DATA
30210 RESTORE 20010:N=0
30220 READ A:IF A<>-1 THEN POKE 1536+N
,A:N=N+1:GOTO 30220
30230 RETURN
32000 DATA 104,160,0,173,145,6,24,105,
```

```
3,133,204,169,0,133,203,145,203,136,20
8,251,230,204,145,203,136,208
32001 DATA 251,230,204,145,203,136,208
,251,230,204,145,203,136,208,251,230,2
04,145,203,136,208,251,162,128,160,57
32002 DATA 169,6,76,92,228,173,150,6,2
40,3,76,95,228,173,0,211,41,15,141,135
,6,162,3,173,10,210
32003 DATA 45,152,6,208,93,169,15,157,
136,6,173,10,210,45,153,6,240,72,189,1
41,6,24,105,8,205,140
32004 DATA 6,144,15,56,233,16,205,140,
6,144,12,169,14,157,136,6,208,5,169,13
,157,136,6,189,129,6
32005 DATA 24,105,2,205,128,6,144,19,5
6,233,4,205,128,6,144,19,189,136,6,41,
11,157,136,6,24,144
32006 DATA 8,189,136,6,41,7,157,136,6,
24,144,8,173,10,210,41,15,157,136,6,20
2,16,152,173,145,6
32007 DATA 24,105,7,133,204,162,4,173,
134,6,224,0,208,3,173,133,6,72,41,240,
74,74,74,74,133,205
32008 DATA 104,41,15,24,125,159,6,157,
159,6,201,16,144,7,230,205,41,15,157,1
59,6,189,135,6,106,72
32009 DATA 176,15,189,140,6,56,229,205
,201,40,176,2,169,40,157,140,6,104,106
,72,176,15,189,140,6,24
32010 DATA 101,205,201,200,144,2,169,2
00,157,140,6,104,106,72,176,15,189,128
,6,56,229,205,201,45,176,2
32011 DATA 169,45,157,128,6,104,106,17
6,15,189,128,6,24,101,205,201,200,144,
2,169,200,157,128,6,202,16
32012 DATA 132,169,6,133,207,162,4,189
,140,6,133,203,173,148,6,224,0,208,3,2
4,105,64,133,206,160,15
32013 DATA 177,206,145,203,136,16,249,
198,204,202,16,225,173,128,6,141,7,208
```

```
,24,105,2,141,6,208,24,105
32014 DATA 2,141,5,208,24,105,2,141,4,
208,162,3,189,129,6,157,0,208,202,16,2
47,173,9,208,24,109
32015 DATA 10,208,240,6,238,146,6,238,
150,6,173,1,208,24,109,2,208,240,3,238
,147,6,234,234,234,206
32016 DATA 151,6,208,16,169,4,141,151,
6,173,148,6,24,105,16,41,48,141,148,6,
173,154,6,201,64,176
32017 DATA 5,160,0,140,155,6,201,112,1
44,5,160,1,140,155,6,173,155,6,208,9,1
73,154,6,24,109,156
32018 DATA 6,208,7,173,154,6,56,237,15
6,6,141,154,6,141,0,210,169,162,141,1,
210,76,95,229
32019 DATA 104,173,167,6,240,107,165,2
0,205,164,6,240,100,141,164,6,206,165,
6,208,8,173,166,6,240,3
32020 DATA 206,166,6,173,166,6,208,79,
173,165,6,201,160,176,72,201,0,208,9,1
69,2,141,157,6,141,150
32021 DATA 6,96,206,168,6,208,54,169,1
0,141,168,6,173,192,2,201,72,240,19,16
9,72,141,192,2,141,193
32022 DATA 2,141,194,2,141,195,2,208,1
7,24,144,167,169,120,141,192,2,141,193
,2,141,194,2,141,195,2
32023 DATA 173,133,6,56,233,2,141,134,
6,173,146,6,240,13,169,3,174,167,6,208
,2,169,0,141,157,6
32024 DATA 96,173,147,6,240,9,169,1,14
1,157,6,141,150,6,96,165,19,205,158,6,
208,191,169,0,133,19
32025 DATA 238,134,6,24,144,181,-1
```

GRAND PRIX

(CASSETTA 16K, DISCO 24K)

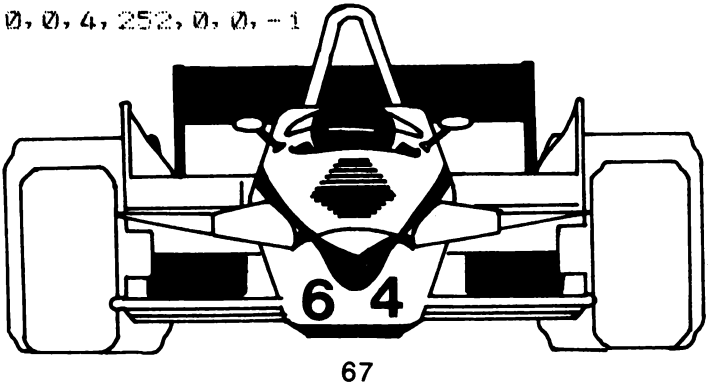
Siete un esperto pilota messo alle strette nella corsa più pericolosa che abbiate mai affrontato. La corsa avviene in 5 livelli, e alla fine di ogni livello la strada si restringe drammaticamente e si fa più tortuosa.
Avete a disposizione cinque automobili; una volta distrutte tutte e cinque le auto, vi sarà mostrato il vostro livello di abilità (il massimo è 100). Se riuscite a portare a termine una corsa ne comincerete una ancora più difficile. Adoperate il joystick in "port 1" per sterzare a destra e a sinistra.

```
0 REM *** GRAND PRIX, BY G. RYAN
10 GOTO 400
20 Z=USR(1540):? TL$;:POKE 85,PEEK(204
)+W:? TR$;:IF NOT Z THEN 20
30 IF Z>255 THEN 200
39 REM NEXT STAGE
40 ST=ST+1:SI=SI+2:W=WD(ST):POKE 1539,
RAND(ST):POKE 77,0:SOUND 1,155-ST*3,2,
6:IF ST=6 THEN 110
50 TL$=L$(SI):TR$=R$(SI):TR$(3)=CHR$(1
57):IF ST>4 THEN 110
60 IF PEEK(204)>10 THEN POSITION 3,0
70 IF PEEK(204)<11 THEN POSITION 30,0
80 POKE 1536,LN(ST):POKE 205,0:POKE 53
278,0
90 ? "Stage ";ST+1;
100 GOTO 20
110 IF ST=6 THEN 130
119 REM FINISHED
120 POKE 1536,17:POKE 205,0:SETCOLOR 4
,10,0:POSITION PEEK(204)+2,0:? "FINISH
":GOTO 20
```

```
130 SOUND 0,0,0,0:SOUND 1,0,0,0:FOR A=
1 TO 200:NEXT A
140 CS=CS+1:IF BN>HS THEN HS=BN
150 ? CHR$(125):POSITION 11,8:? "Cours
e ";CS;" Completed":POSITION 15,10:? "
Crashes ";CR:SOUND 1,255,2,4
160 POKE 53248,120:POSITION 12,13:? "S
kill Rating ";BN:POSITION 8,15:? "High
est Skill Rating ";HS
170 POSITION 2,17:? "Hit START when re
ady for next course"
180 IF PEEK(53279)<>6 AND STRIG(0) THE
N 180
190 WD(0)=10:WD(1)=9:RAND(4)=45:IF ST>
2 THEN WD(2)=8
195 GOTO 680
199 REM CRASHED
200 SOUND 1,0,0,0:FOR A=1 TO 15:NEXT A
:PM$(IN+YP)=E$
210 FOR A=15 TO 0 STEP -0.5:SOUND 0,RN
D(0)*50+100,8,A:POKE 704,48+A:NEXT A
220 CR=CR+1:BN=BN-(6-ST)*(CR+1):BN=BN*
(BN>0)
230 PM$(2)=PM$:POKE 704,CC:IF CR<5 THE
N SI=SI-2:ST=ST-1:GOTO 690
239 REM GAME OVER
240 POSITION 15,10:? "GAME OVER":POSIT
ION 13,4:BN=INT(ST*4+BN/50+0.5)
250 IF BN>HS THEN HS=BN
260 ? "Skill Rating ";BN:POSITION 9,7:
? "Highest Skill Rating ";HS:POSITION
15,13:? "Hit START":CS=0
270 IF PEEK(53279)<>6 AND STRIG(0) THE
N 270
280 GOTO 680
400 GRAPHICS 18:POSITION 5,4:? #6;"gra
nd prix"
410 POSITION 1,6:? #6;"WRITTEN BY G. R
YAN"
```

```
420 FOR A=53248 TO 53265:POKE A,0:NEXT
 A
499 REM INITIALIZING
500 DIM PM$(1200),C$(24),E$(24),TL$(2)
,TR$(3),L$(13),R$(13)
510 PM$(1)=CHR$(0):PM$(1200)=CHR$(0):P
M$(2)=PM$
520 RESTORE 5000:TRAP 540:FOR A=1540 T
O 1777:READ D:POKE A,D:NEXT A
540 DIM WD(6),RAND(6),LN(6)
560 A=ADR(PM$):D=INT(A/1024)*1024+512:
IF D<A THEN D=D+1024
570 IN=D-A:POKE 559,46:POKE 53277,3:PO
KE 54279,(D-512)/256
580 RESTORE 590:FOR A=0 TO 5:READ B,C,
D:WD(A)=B:RAND(A)=C:LN(A)=D:NEXT A
587 REM DATA FOR TRACK WIDTH, DIRECTIO
N CHANGES, LENGTH OF STAGE
590 DATA 11,45,210,10,50,230,9,55,220,
9,55,220,8,30,120,8,20,20
600 RESTORE 620:FOR A=1 TO 12:READ D:L
$(A)=CHR$(D):NEXT A
610 FOR A=1 TO 12:READ D:R$(A)=CHR$(D)
:NEXT A
620 DATA 19,19,20,20,9,12,34,34,46,46,
19,19
630 DATA 19,19,20,20,11,15,34,34,46,46
,19,19
640 RESTORE 650:FOR A=1 TO 24:READ D:C
$(A)=CHR$(D):NEXT A
650 DATA 0,0,0,0,0,0,24,90,126,90,24,6
0,60,60,219,255,219,0,0,0,0,0,0,0
660 RESTORE 670:FOR A=1 TO 24:READ D:E
$(A)=CHR$(D):NEXT A
665 POSITION 4,10:? #6;"PRESS START...
"
667 IF PEEK(53279)<>6 AND STRIG(0) THE
N 667
670 DATA 0,0,0,0,0,16,66,8,162,9,84,36
,82,9,130,32,5,80,0,0,0,0,0,0,0
```

```
675 POKE 82,0:POKE 83,39:GRAPHICS 0:PO
KE 710,0:POKE 709,14:POKE 752,1:POKE 5
59,46:POKE 53277,3:? " "
680 ST=-1:BN=100:SI=-1:CR=0:CC=INT(RND
(0)*15)*16+10:POKE 1536,2:POKE 1537,2:
POKE 1538,24:POKE 1539,20:YP=85
690 POKE 204,14:POKE 205,0:POKE 203,12
0:POKE 1536,1:POKE 712,0:Z=ST:IF Z<0 T
HEN Z=0
700 B=Z*2+1:? CHR$(125):FOR A=0 TO 23:
POSITION 14,A:? L$(B,B+1);:POSITION 14
+WD(Z),A:? R$(B,B+1);:NEXT A
710 PM$(IN+YP)=C$:POKE 704,CC:POKE 532
48,120:POKE 53278,0
740 GOTO 40
4999 REM DATA FOR USR SUBROUTINE
5000 DATA 104,173,120,2,74,74,41,3,170
,189,115,6,141,0,210,189,119,6,141,1,2
10,165,203,24,125,123,6
5010 DATA 133,203,141,0,208,160,0,132,
212,132,84,173,4,208,133,213,206,0,6,2
08,2,198,212,165,204,166,205
5020 DATA 24,125,111,6,133,204,133,85,
205,1,6,240,9,205,2,6,208,10,169,1,208
,2,169,2,133,205,208
5030 DATA 16,173,10,210,205,3,6,176,8,
41,7,170,189,103,6,133,205,96,0,1,2,0,
1,2,1,2,0
5040 DATA 255,1,0,0,27,27,0,0,132,132,
0,0,4,252,0,0,-1
```



ESCAPE

(CASSETTA 16K, DISCO 24K)

Tornando sulla Terra a bordo della vostra navicella spaziale vi trovate di fronte ad un problema quasi insormontabile: sul terreno sono state disseminate quasi dappertutto delle mine ed un muro giallo ostacola la vostra pista di atterraggio. Dovete schivare tutte le mine ed imbucare la stretta apertura nel muro se desiderate rivedere i vostri familiari.
Premete il bottone FIRE per far muovere la navetta sullo schermo e adoperate un joystick in "port 1" per spostarla verso l'alto o verso il basso. Se riuscite a fuggire, altre mine verranno disseminate e dovrete nuovamente premere FIRE. Le capsule fra le mine vi fanno guadagnare punti.

```
0 REM ESCAPE, written by G. Ryan
10 GOTO 500
20 S=STICK(0):YP=YP+(S=13)-(S=14):XP=X
P+1:POKE 53248,XP:PM$(YP)=PL$:IF  NOT
PEEK(53252) THEN 20
30 IF PEEK(53252)<>4 THEN 200
39 REM PLAYER HIT SOMETHING
40 X=INT((XP-48)/8):Y=INT((YP-IN-13)/4
):IF XP>176 THEN 650
50 FOR A=N TO N1:FOR B=-N1 TO N2:LOCAT
E X+A,Y+B,C:IF C<>164 THEN NEXT B:NEXT
 A:POKE 53278,N:GOTO 20
60 POP :POP :SC=SC+ST*INT(RND(N)*N2+N1
):POSITION 7,23:? #N6;SC
70 COLOR 32:PLOT X+A,Y+B:POKE 53278,N
80 FOR A=N1 TO 15:SOUND N,PEEK(53770),
10,10:NEXT A:SOUND N,N,N,N
90 POKE 53278,N:GOTO 20
100 FOR A=N TO 511:POKE D+A,PEEK(57344
+A):NEXT A:RETURN
```

```
200 FOR A=15 TO N STEP -N3:FOR D=N TO
N2
210 SOUND D,RND(N)*50+100,8,A:NEXT D:P
OKE 704,32+A:NEXT A:POKE 704,N
220 POSITION 7,23:? #N6;SC:LV=LV-N1:IF
 LV>N THEN 290
230 POSITION N3,10:? #N6;"  GAME OVER
 ":IF SC>HS THEN HS=SC
250 POSITION N2,13:? #N6;"  hi score!
 ";HS:POSITION N3,17:? #N6;"  HIT star
t"
260 IF PEEK(53279)<>N6 AND STRIG(N) TH
EN 260
270 POKE 77,N:GOTO 580
290 XP=38:YP=IN+61:POKE 53248,XP:POKE
704,40:POKE 53278,N:PM$(N2)=PM$
300 IF STRIG(N) THEN 300
310 GOTO 20
499 REM INITIALIZING
500 READ N,N1,N2,N3,N6,E,W:DATA 0,1,2,
3,6,255,256
510 A=PEEK(740)-4:POKE 106,A-2:GRAPHIC
S 18:FOR X=53248 TO 53255:POKE X,N:NEX
T X
520 POSITION 7,4:? #N6;"escape":POSITI
ON N1,N6:? #N6;"WRITTEN BY G. RYAN"
530 D=A*W:GOSUB 100:RESTORE 5000
540 READ A:IF A<>-N1 THEN FOR X=A+D TO
 A+D+7:READ A:POKE X,A:NEXT X:GOTO 540
550 DIM PM$(1200),PL$(15):PM$(N1)=CHR$
(N):PM$(1200)=PM$(N1):PM$(N2)=PM$
560 RESTORE 5500:FOR A=N1 TO 13:READ X
:PL$(A)=CHR$(X):NEXT A
570 A=ADR(PM$):PM=INT(A/1024)*1024:X=P
M+512:IN=X-A:IF A>X THEN PM=PM+1024:X=
X+1024:IN=IN+1024
580 YP=IN+60:GRAPHICS 17:POKE 54279,PM
/W:POKE 53277,N3:POKE 559,46
590 POKE 704,40:POKE 623,N1:POKE 756,P
```

```
EEK(106)+N2
600 SETCOLOR N,8,N6:SETCOLOR N1,N3,8:S
ETCOLOR N2,11,8
610 SETCOLOR N3,14,10
620 COLOR 35:PLOT N,N:DRAWTO 19,N:PLOT
 N,22:DRAWTO 19,22
630 COLOR 129:PLOT 18,N1:DRAWTO 18,21:
COLOR 165:PLOT 19,N1:DRAWTO 19,21
640 POSITION N1,23:? #N6;"score 0   st
age 1":LV=3:ST=N:SC=N:G=N1:M=N
650 SC=SC+M:YP=IN+61:ST=ST+1:POSITION
17,23:? #N6;ST:POSITION 7,23:? #N6;SC:
PM$(N2)=PM$
660 M=INT(RND(N)*N3)+N1:FOR A=N1 TO M:
COLOR N2:PLOT RND(N)*12+N2,RND(N)*20+N
1:COLOR 32
670 PLOT RND(N)*14+N2,RND(N)*20+N1:PLO
T RND(N)*14+N2,RND(N)*20+N1:NEXT A:COL
OR 164
680 PLOT RND(N)*12+N3,RND(N)*20+N1:COL
OR 129:PLOT 18,G:DRAWTO 18,G+N2:G=INT(
RND(N)*17+N1):COLOR 32
690 PLOT 18,G:DRAWTO 18,G+N2:YP=IN+60:
XP=38:POKE 53278,N
700 IF STRIG(N) THEN 700
710 POKE 53278,N:GOTO 20
4999 REM DATA FOR NEW CHARACTERS AND P
LAYERS SHIP
5000 DATA 8,238,238,238,14,238,238,238
,224
5010 DATA 24,0,247,247,247,0,127,127,1
27
5020 DATA 16,129,90,36,90,90,36,90,129

5030 DATA 32,0,16,56,84,254,84,56,16
5040 DATA 40,0,64,0,2,0,16,0,0
5050 DATA -1
5500 DATA 0,0,0,112,224,88,239,88,224,
112,0,0,0
```

TIME TYPIST

(CASSETTA 16K, DISCO 24K)

In questo gioco il computer mostrerà una sequenza casuale di 20 lettere. Dovrete ribatterle nello stesso ordine nel minor tempo possibile; il timer, nel frattempo, si avvierà verso lo zero. Premete BACK SPACE se fate un errore. Alla fine della serie un'altra sequenza di lettere viene visualizzata ed avete ancor meno tempo per ripeterla. È un gioco eccellente per chiunque voglia far pratica di battitura a macchina e voglia seguirne i progressi.

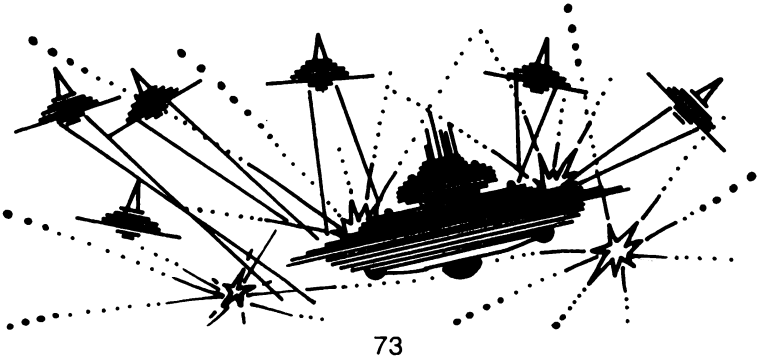
```
10 REM **TIME TYPIST
20 REM **By Cliff McConnell
110 DIM L$(30),P$(30):OPEN #1,4,0,"K"
120 GRAPHICS 18:SC=0:HS=SC
150 REM
210 POSITION 1,0:? #6;"SCORE:";SC;"
"
220 POSITION 1,1:? #6;"HIGH SCORE:";HS

230 POSITION 1,5:? #6;"please press st
art"
240 IF PEEK(53279)<>6 THEN 240
245 POSITION 1,5:? #6;"
   "
250 ST=1
270 T=15+10*(ST=1)-(ST>2)*(ST-2)
280 N=20
310 X=INT((20-N)/2):P=0
510 GOSUB 10010:POSITION 1,2:? #6;"TIM
E:";T
710 POKE 20,0:POKE 19,0:POKE 20,0
720 IF PEEK(20)>50 THEN POKE 20,0:T=T-
1
```

```
730 POSITION 6,2:? #6;T;" ";:IF  NOT T
 THEN 8010
750 IF PEEK(764)=255 THEN 720
760 GET #1,A:POKE 764,255:IF A<65 OR A
>90 THEN 1010
770 IF P<N THEN POSITION X+P,7:? #6;CH
R$(A+128):P=P+1:P$(P,P)=CHR$(A):IF P$=
L$ THEN 5010
780 GOTO 720
1010 IF A=126 AND P>0 THEN P=P-1:POSIT
ION X+P,7:? #6;" ":P$(P+1,P+1)=" ":GOT
O 720
1020 IF A=32 AND P<N THEN P=P+1:GOTO 7
20
2010 GOTO 720
5000 REM **DID IT!!
5010 SC=SC+ST*T:POSITION 7,0:? #6;SC
5090 ST=ST+1:GOTO 270
8000 REM **DIDN'T MAKE IT
8010 POSITION 5,4:? #6;"bad luck":FOR
K=0 TO 200:NEXT K:POSITION 5,4:? #6;"
      "
8020 IF SC>HS THEN HS=SC
8050 GOTO 150
10000 REM **PRINT UP LETTERS
10010 POSITION 5,4:? #6;"get ready"
10015 POSITION 0,7:? #6;"
       ":POSITION 0,6:? #6;"
         "
10018 POSITION INT((20-N)/2),6
10020 P$="":L$="":FOR K=1 TO N:A=INT(R
ND(0)*26)+65:? #6;CHR$(A+160);:L$(K,K)
=CHR$(A)
10030 SOUND 0,100,10,10:FOR G=0 TO 15:
NEXT G:SOUND 0,0,0,0:NEXT K
10050 POSITION 5,4:? #6;"          "
10110 RETURN
```

GALAX ATTACK

(CASSETTA 24K, DISCO 32K)

Scopo del gioco è la distruzione di ondate successive di alieni in attacco. Questi spiacevoli alieni si staccano dalla loro formazione gettandosi in planata sulla vostra nave collocata in basso sullo schermo, cercando di bombardarla prima di allontanarsi.
La distruzione di un mutante (un alieno multicolore che cambia aspetto dopo che ha abbandonato la formazione) vi farà ottenere dei bonus a seconda del colore dell'alieno che avrete colpito prima del mutante. La distruzione di altri alieni vi farà guadagnare fra i dieci ed i quaranta punti.
Adoperate il joystick in "port 1" per muovere la vostra nave da destra a sinistra. Premete il pulsante FIRE per lanciare un missile. Dopo una pausa, mentre il programma si avvia in linguaggio macchina con il POKE, apparirà sullo schermo la videata iniziale del gioco.
Premete OPTION per selezionare il numero di bombe che gli alieni possono avere a disposizione. Premete SELECT per decidere il livello di gioco e START per iniziare la partita. Dopo che sarete stati distrutti per tre volte, il programma ritornerà all'inizio e vi indicherà il punteggio ottenuto.



```
10 REM **GALAX ATTACK
20 REM **By Cliff McConnell
30 REM **ALL REMS MAY BE OMITTED
90 FOR T=1536 TO 1536+255:POKE T,0:NEX
T T
110 TOP=PEEK(740):POKE 106,TOP-5:POKE
0,TOP-8:START=(TOP-16)*256:CH=(TOP-18)
*256:POKE 1769,CH/256
120 DVBLANK=START:IVBLANK=START+1223:D
LI=START+1790:USER=START+1802:CLEAR=ST
ART+1822
125 GRAPHICS 18:DL=PEEK(560)+256*PEEK(
561):SCRN=PEEK(88)+256*PEEK(89):FOR G=
1 TO 11
127 TRAP 129:? #6;"*****************
*";:NEXT G
128 ? #6;"****":GOTO 130
129 ? #6:GOTO 127
130 POSITION 9,4:? #6;"galax":POSITION
11,6:? #6;"attack"
135 POKE DL+9,PEEK(DL+11)+16:POKE 5427
6,4
140 N0=0:N1=1:N2=2:N6=6:GOTO 5010
200 REM **DATA FOR NEW CHARACTER SET
210 DATA 0,0,0,0,0,0,0,0,0,126,219,255
,129,195,126,0,8,28,62,62,127,127,127,
255,4,14
211 DATA 31,31,63,63,63,127,2,7,15,15,
31,31,31,63,1,3,7,7,15,15,15,31,0,1,3,
3
212 DATA 7,7,7,15,0,0,1,1,3,3,3,7,0,0,
0,0,1,1,1,3,0,0,0,0,0,0
213 DATA 0,1,0,0,0,0,0,0,0,128,0,0,0,0
,128,128,128,192,0,0,128,128,192,192,1
92,224
214 DATA 0,128,192,192,224,224,224,240
,128,192,224,224,240,240,240,248,64,22
4,240,240,248,248,248,252,32,112
215 DATA 248,248,252,252,252,254,16,56
,124,124,254,254,254,255,0,0,0,0,8,8,8
```

```
,8,0,0,0,0
216 DATA 4,4,4,4,0,0,0,0,2,2,2,2,0,0,0
,0,1,1,1,1,0,0,0,0,128,128
217 DATA 128,128,0,0,0,0,64,64,64,64,0
,0,0,0,32,32,32,32,0,0,0,0,16,16,16,16

218 DATA 8,8,8,8,0,0,0,0,4,4,4,4,0,0,0
,0,2,2,2,2,0,0,0,0,1,1
219 DATA 1,1,0,0,0,0,128,128,128,128,0
,0,0,0,64,64,64,64,0,0,0,0,32,32,32,32

220 DATA 0,0,0,0,16,16,16,16,0,0,0,0,1
53,90,60,255,255,60,90,153,0,127,65,65
,65,65
221 DATA 65,127,0,0,62,34,34,34,62,0,0
,0,0,28,28,28,0,0,147,153,205,103,51,2
7,15,3,201,153,179,230,204,216
222 DATA 240,192,-1
2200 REM **MACHINE LANGUAGE DATA
2210 DATA 173,156,6,240,3,76,98,228,17
3,0,211,106,106,106,176,35,173,133,6,2
01,221,144,66,206,132,6
2211 DATA 16,61,169,7,141,132,6,169,0,
172,133,6,200,145,88,172,131,6,145,88,
206,133,6,208,38,106
2212 DATA 176,35,173,133,6,201,239,176
,28,238,132,6,173,132,6,41,7,141,132,6
,208,15,169,0,172,133
2213 DATA 6,145,88,172,131,6,145,88,23
8,133,6,173,142,6,208,58,173,133,6,56,
233,20,174,132,6,168
2214 DATA 169,0,224,4,144,9,145,88,200
,208,8,240,2,240,226,200,145,88,136,14
0,131,6,173,132,6,141
2215 DATA 130,6,173,16,208,205,242,6,2
40,105,141,242,6,201,1,240,98,169,1,14
1,142,6,173,130,6,73
2216 DATA 8,141,130,6,41,8,208,81,173,
131,6,56,233,20,141,131,6,168,177,88,4
1,63,201,1,208,43
```

```
2217 DATA 177,88,24,105,64,201,64,176,
26,162,0,189,193,6,240,3,232,208,248,1
52,157,193,6,169,200,141
2218 DATA 244,6,169,4,157,203,6,169,0,
145,88,173,131,6,24,144,7,173,131,6,20
1,21,176,13,24,105
2219 DATA 20,168,169,0,145,88,141,142,
6,240,128,173,231,6,240,11,206,231,6,2
08,17,173,230,6,141,229
2220 DATA 6,206,229,6,208,6,173,232,6,
141,231,6,160,4,185,115,6,208,54,173,2
31,6,208,84,173,10
2221 DATA 210,201,150,176,77,133,203,1
52,72,164,203,177,88,133,204,41,63,201
,1,240,5,104,168,24,144,56
2222 DATA 169,0,145,88,104,168,165,203
,141,164,6,162,0,134,205,173,164,6,56,
176,2,208,102,233,20,201
2223 DATA 20,144,15,141,164,6,165,205,
24,105,20,133,205,232,224,6,144,227,18
9,16,6,153,115,6,173,164
2224 DATA 6,56,176,3,56,176,115,233,20
,74,170,189,24,6,153,120,6,238,157,6,1
73,10,210,41,15,153
2225 DATA 125,6,165,203,24,144,3,24,14
4,130,153,145,6,165,204,153,165,6,41,1
92,74,74,74,74,74,74
2226 DATA 170,189,196,2,192,0,240,3,15
3,191,2,169,100,141,243,6,169,166,141,
1,210,173,10,210,208,21
2227 DATA 173,10,210,41,1,208,8,185,12
5,6,41,254,24,144,3,25,125,6,153,125,6
,185,125,6,74,170
2228 DATA 189,134,6,41,248,74,74,74,13
3,203,189,134,6,41,7,24,144,3,24,144,3
3,121,150,6,153,150
2229 DATA 6,201,8,144,13,24,144,3,24,1
44,148,41,7,153,150,6,230,203,185,125,
6,41,1,208,24,24
2230 DATA 144,3,24,144,63,185,120,6,24
```

```
,101,203,201,200,144,29,185,125,6,9,1,
153,125,6,185,120,6
2231 DATA 56,229,203,201,45,176,11,185
,125,6,41,254,153,125,6,24,144,217,153
,120,6,173,144,6,41,248
2232 DATA 74,74,74,133,203,173,144,6,4
1,7,24,144,6,24,144,170,24,144,95,121,
159,6,153,159,6,201
2233 DATA 8,144,7,41,7,153,159,6,230,2
03,185,115,6,24,101,203,201,230,144,65
,152,72,170,188,115,6
2234 DATA 162,15,24,105,3,24,101,0,133
,204,169,0,133,203,145,203,136,202,16,
250,104,168,72,185,165,6
2235 DATA 170,185,145,6,168,138,145,88
,104,168,169,0,153,120,6,153,115,6,206
,157,6,192,0,208,5,169
2236 DATA 4,141,114,6,24,144,3,153,115
,6,136,16,152,206,191,6,208,72,169,2,1
41,191,6,162,9,189
2237 DATA 180,6,240,30,189,170,6,73,8,
157,170,6,240,20,189,180,6,24,105,20,1
57,180,6,208,9,222
2238 DATA 180,6,160,236,169,0,145,88,2
02,16,218,173,227,6,240,11,206,227,6,2
08,17,173,226,6,141,225
2239 DATA 6,206,225,6,208,6,173,228,6,
141,227,6,173,10,210,41,3,168,185,115,
6,240,56,132,203,173
2240 DATA 190,6,201,9,176,43,173,227,6
,208,67,173,10,210,41,3,208,60,162,0,1
89,180,6,240,5,232
2241 DATA 224,9,144,246,173,10,210,41,
3,208,43,173,10,210,41,255,168,192,20,
144,87,192,240,176,83,144
2242 DATA 2,240,79,177,88,41,63,201,1,
208,71,152,24,105,20,157,180,6,238,190
,6,169,8,157,170,6
2243 DATA 208,54,164,203,185,115,6,56,
233,48,74,74,74,74,168,185,34,6,133,20
```

```
4, 164, 203, 185, 120, 6, 56
2244 DATA 233, 44, 74, 74, 74, 24, 101, 204, 2
4, 105, 20, 201, 190, 176, 15, 201, 40, 144, 11,
157, 180, 6, 169, 8, 157, 170
2245 DATA 6, 238, 190, 6, 162, 4, 189, 115, 6,
240, 39, 189, 214, 6, 208, 34, 173, 142, 6, 73, 1
, 208, 52, 189, 235, 6
2246 DATA 240, 78, 189, 115, 6, 56, 233, 52, 7
4, 74, 74, 74, 168, 185, 34, 6, 133, 203, 189, 12
0, 6, 56, 176, 3, 24, 144
2247 DATA 71, 233, 48, 74, 74, 74, 24, 101, 20
3, 168, 204, 131, 6, 208, 4, 240, 8, 16, 191, 200
, 204, 131, 6, 208, 64, 189
2248 DATA 214, 6, 208, 59, 169, 200, 141, 244
, 6, 169, 1, 157, 214, 6, 169, 4, 157, 219, 6, 172
, 131, 6, 173, 133, 6, 56
2249 DATA 176, 2, 240, 122, 233, 20, 141, 131
, 6, 169, 0, 145, 88, 141, 142, 6, 152, 24, 144, 4
, 144, 119, 16, 197, 105, 20
2250 DATA 168, 169, 0, 145, 88, 224, 0, 208, 5
5, 240, 2, 208, 89, 172, 114, 6, 185, 109, 6, 133
, 203, 152, 10, 10, 10, 10
2251 DATA 10, 10, 24, 105, 16, 160, 12, 145, 8
8, 136, 145, 88, 136, 24, 144, 2, 16, 208, 101, 2
03, 145, 88, 169, 60, 141, 192
2252 DATA 6, 169, 4, 141, 114, 6, 165, 203, 16
0, 17, 208, 17, 189, 165, 6, 74, 74, 74, 74, 74, 7
4, 141, 114, 6, 24, 105
2253 DATA 1, 160, 18, 24, 113, 88, 201, 26, 14
4, 10, 56, 233, 10, 145, 88, 136, 169, 1, 208, 23
9, 145, 88, 240, 53, 173, 133
2254 DATA 6, 56, 233, 220, 133, 203, 189, 115
, 6, 201, 220, 144, 38, 189, 120, 6, 56, 233, 44,
56, 237, 132, 6, 74, 74, 74
2255 DATA 197, 203, 208, 21, 234, 234, 234, 2
34, 234, 234, 234, 169, 1, 141, 155, 6, 141, 156
, 6, 141, 224, 6, 76, 98, 228, 202
2256 DATA 16, 140, 173, 192, 6, 240, 17, 206,
192, 6, 208, 12, 169, 0, 160, 10, 145, 88, 200, 1
45, 88, 200, 145, 88, 76, 98
```

```
2257 DATA 228,173,155,6,240,3,76,95,22
8,162,4,189,3,208,157,235,6,202,208,24
7,162,3,169,0,24,125
2258 DATA 0,208,202,16,249,141,235,6,1
72,133,6,177,88,201,18,144,31,201,26,1
76,27,56,233,18,56,233
2259 DATA 4,41,7,205,132,6,240,2,144,1
2,238,224,6,238,155,6,238,156,6,76,95,
228,200,177,88,201
2260 DATA 18,144,31,201,26,176,27,56,2
33,18,56,233,4,41,7,205,132,6,240,2,17
6,12,238,224,6,238
2261 DATA 155,6,238,156,6,76,95,228,20
6,240,6,208,8,169,3,141,240,6,141,30,2
08,165,0,24,105,7
2262 DATA 133,204,169,0,133,203,162,4,
188,115,6,240,20,189,214,6,208,15,138,
72,162,15,189,0,6,145
2263 DATA 203,136,202,16,247,104,170,1
98,204,202,16,226,162,4,189,214,6,240,
106,138,72,189,219,6,10,10
2264 DATA 10,10,170,104,72,24,101,0,24
,105,3,133,204,169,0,133,203,104,72,17
0,188,115,6,189,219,6
2265 DATA 208,36,162,25,169,0,145,203,
136,202,16,250,104,170,206,234,6,208,9
,238,155,6,238,156,6,76
2266 DATA 95,228,169,0,157,115,6,157,2
14,6,240,35,169,15,133,205,138,24,105,
15,170,189,45,6,145,203
2267 DATA 202,136,198,205,16,245,104,1
70,206,245,6,208,8,169,2,141,245,6,222
,219,6,202,16,142,172,133
2268 DATA 6,173,132,6,24,105,2,145,88,
24,105,8,200,145,88,173,142,6,240,11,1
73,131,6,24,105,20
2269 DATA 168,169,0,145,88,172,131,6,1
73,130,6,24,105,18,24,105,64,145,88,16
2,4,189,120,6,157,255
2270 DATA 207,202,208,247,173,120,6,14
```

```
1,7,208,24,105,2,141,6,208,24,105,2,14
1,5,208,24,105,2,141
2271 DATA 4,208,162,9,188,180,6,240,93
,192,240,176,20,192,20,144,16,177,88,2
40,6,41,63,201,1,240
2272 DATA 6,189,170,6,24,144,44,188,18
0,6,177,88,41,63,201,1,240,4,169,0,145
,88,189,180,6,56
2273 DATA 233,20,168,177,88,41,63,201,
1,240,4,169,0,145,88,206,190,6,169,0,1
57,180,6,240,25,188
2274 DATA 180,6,24,105,18,145,88,152,5
6,233,20,168,177,88,41,63,201,1,240,4,
169,0,145,88,202,16
2275 DATA 155,206,213,6,208,91,169,2,1
41,213,6,162,9,188,193,6,240,34,189,20
3,6,208,21,169,0,157
2276 DATA 193,6,206,234,6,208,9,238,15
5,6,238,156,6,76,95,228,169,223,24,105
,33,145,88,222,203,6
2277 DATA 202,16,214,173,243,6,240,13,
24,105,5,144,2,169,0,141,243,6,141,0,2
10,173,244,6,240,19
2278 DATA 162,136,56,233,10,80,3,169,0
,170,141,244,6,141,2,210,142,3,210,76,
95,228,72,173,233,6
2279 DATA 141,10,212,141,9,212,104,64,
104,169,0,141,155,6,141,156,6,173,224,
6,208,5,173,234,6,208
2280 DATA 246,96,104,165,0,24,105,3,13
3,205,169,0,133,204,162,5,160,0,152,14
5,204,136,208,251,230,205
2281 DATA 202,208,246,96,0,-1
3200 REM **PAGE SIX DATA
3210 DATA 0,0,0,24,60,126,219,219,255,
195,102,60,24,0,0,0,56,72,98,104,102,1
18,134,150,58,74
3211 DATA 90,106,122,138,154,170,186,2
02,0,20,40,60,80,100,120,140,160,180,2
00,0,0,0,42,42,28,28
```

```
3212 DATA 127,127,28,28,42,42,0,0,0,0,
0,0,0,62,62,34,34,34,34,62,62,0,0,0,0,
0
3213 DATA 0,0,0,0,28,28,20,20,28,28,0,
0,0,0,0,0,0,0,0,0,0,28,28,28,28,0
3214 DATA 0,0,0,0,0,1,2,4,8,0,0,-1
5000 REM **POKE IN CHARACTERS
5010 POSITION 8,7:? #N6;"PLEASE**WAIT"
:TRAP 5110:FOR N=CH TO CH+6666:READ A:
POKE N,A:NEXT N
5100 REM **POKE IN ML
5110 POSITION 7,7:? #N6;"NOT*LONG*TO*G
O":TRAP 5210:FOR N=START TO START+6666
:READ A:POKE N,A:NEXT N
5200 REM **POKE IN PAGE 6 DATA
5210 POSITION 6,7:? #N6;"NEARLY**FINIS
HED":TRAP 5310:FOR N=1536 TO 9999:READ
 A:POKE N,A:NEXT N
5300 REM
5310 POKE 1749,N1:POKE 1727,N1:POKE 17
63,N1:POKE 1767,N1:POKE 1691,N1:POKE 1
692,N1:SC=N0:HS=N0:LEVEL=N1:BO=9
5410 GRAPHICS 18:POKE 53277,3:POKE 559
,62:POKE 54279,TOP-8:POKE 1781,2
5411 POSITION N0,N0:? #N6;"high";HS;"
          ";:POSITION 10,N0:? #N6;"scor
e";SC;"      "
5412 POSITION N0,N1:? #N6;"press:":? #
N6;"start TO START GAME"
5414 ? #N6;"select LEVEL:";LEVEL
5416 ? #N6;"option BOMBS:";BO
5420 IF PEEK(53279)=6 THEN 10001
5430 IF PEEK(53279)=5 THEN 6010
5440 IF PEEK(53279)<>3 THEN 5420
5450 REM **PRESSED OPTION
5460 BO=(BO+N1)*(BO<9):POSITION 13,4:?
 #N6;BO
5470 IF PEEK(53279)=3 THEN 5470
5480 GOTO 5420
6000 REM **PRESSED SELECT
```

```
6010 LEVEL=LEVEL*(LEVEL(9)+1:POSITION
13,3:? #N6;LEVEL;" "
6020 IF PEEK(53279)=5 THEN 6020
6030 GOTO 5420
10000 REM **START GAME
10001 POKE 0,TOP-8:A=USR(CLEAR)
10005 ? #N6;CHR$(125):POSITION 15,N0:?
 #N6;"00000"
10007 POKE 54286,N0:HI=INT(IVBLANK/256
):POKE 547,HI:POKE 546,IVBLANK-HI*256:
HI=INT(DVBLANK/256):POKE 549,HI
10008 POKE 548,DVBLANK-HI*256:HI=INT(D
LI/256):POKE 513,HI:POKE 512,DLI-HI*25
6:POKE 54286,192
10009 POKE DL+3,PEEK(DL+3)+128
10010 SC=N0:SHIPS=N2+N1:STAGE=LEVEL
11010 ALIENO=20+STAGE*N2:K=8:N=0.8+STA
GE*0.2:IF N>3 THEN N=3
11020 IF K>16 THEN K=16
11030 IF ALIENO>40 THEN ALIENO=40
11040 POKE 1680,K:POKE 1770,ALIENO:FOR
 K=0 TO 7:POKE 1670+K,K*N:NEXT K
11050 POKE 1762,20:POKE 1764,20:POKE 1
766,100:POKE 1768,20
11060 N=INT((ALIENO-N1)/10):K=ALIENO-N
*10:X=(21-(K*N2))/N2:POSITION X,N+N1:F
OR Y=1 TO K:? #N6;"! ";:NEXT Y
11070 POSITION N1,N1:FOR Y=N1 TO ALIEN
O-K:? #N6;"! ";
11075 NEXT Y:POSITION N0,N0:? #N6;"sta
ge";STAGE;"     ":FOR K=N1 TO 200:NEXT
K:POSITION N0,N0:? #N6;"          "
11077 FOR T=N0 TO 4:POKE 1750+T,N0:POK
E 1755+T,N0:NEXT T
11080 FOR K=N0 TO 4:POKE 1771+K,N0:NEX
T K:POKE 53278,N0:POKE 1760,N0
11081 POKE 1767,50:POKE 77,N0
11082 POKE 712,0:POKE 711,56:POKE 710,
200:POKE 709,138:POKE 708,40
```

```
11090 A=SCRN+PEEK(1667):IF PEEK(A)>81
AND PEEK(A)<99 THEN POKE A,N0
11091 IF PEEK(A+20)>81 AND PEEK(A+20)<
99 THEN POKE A+20,N0
11093 FOR K=N0 TO 9:N=SCRN+PEEK(1716+K
):IF PEEK(N)>17 AND PEEK(N)<35 THEN PO
KE N,N0
11095 IF PEEK(N+20)>17 AND PEEK(N+20)<
35 THEN POKE N+20,N0
11097 NEXT K:POKE SCRN+PEEK(1669),N0:P
OKE SCRN+PEEK(1669)+1,N0
11098 FOR G=0 TO 4:IF PEEK(1651+G)>20
THEN A=(TOP-5+G)*256+PEEK(1651+G):FOR
K=A TO A-20 STEP -1:POKE K,N0:NEXT K
11099 NEXT G:FOR T=N0 TO 4:POKE 1651+T
,N0:NEXT T:POKE 1669,230:POKE 1667,210
12000 REM
12010 FOR K=N0 TO 9:POKE 1716+K,N0:NEX
T K:POKE 1726,9-B0
12011 POKE 1691,N1:POKE 1692,N1:POKE 1
778,N0:POKE 1760,N0
12016 TRAP 12020:WA=USR(USER):SOUND N0
,N0,N0,N0:SOUND N1,N0,N0,N0
12020 IF PEEK(1760)>N0 THEN 15010
12030 STAGE=STAGE+N1:GOTO 11010
15000 REM **PLAYER DIED
15010 SHIPS=SHIPS-N1:POSITION N0,N0:?
#N6;"SHIPS LEFT:";SHIPS;" ";
15015 POKE SCRN+PEEK(1669),38:POKE SCR
N+PEEK(1669)+N1,39:FOR T=200 TO N0 STE
P -3:SOUND N0,T,6,10:NEXT T
15020 POKE SCRN+PEEK(1669),N0:POKE SCR
N+PEEK(1669)+N1,N0:SOUND N0,N0,N0,N0
15045 FOR K=N0 TO 4:IF PEEK(1651+K)>N0
 THEN POKE SCRN+PEEK(1681+K),PEEK(1701
+K)
15046 NEXT K
15048 POSITION N0,N0:? #N6;"
  ";
15049 SC=PEEK(SCRN+18)-16+(PEEK(SCRN+1
```

```
7)-16)*10+(PEEK(SCRN+16)-16)*100+(PEEK
(SCRN+15)-16)*1000
15050 FOR K=N0 TO 4:IF PEEK(1750+K)>N0
 THEN POKE 1750+K,N0:POKE 1770,PEEK(17
70)-N1
15051 NEXT K:POKE 1693,N0:IF PEEK(1729
)=1 AND PEEK(1739)<2 THEN POKE 1739,2
15053 SC=SC*10:IF SHIPS>N0 THEN 11080
15055 IF SC>HS THEN HS=SC
15060 POSITION N0,N0:? #N6;"        game
 over      ":FOR K=N1 TO 120:NEXT K:GOT
O 5410
```

CHOPPER MISSION

(CASSETTA 24K, DISCO 32K)

Degli ostaggi sono stati buttati fuori da un aereo in volo ma fortunatamente è stato messo in funzione un raggio antigravitazionale per rallentare la loro velocità di caduta; purtroppo altre minacce attentano ancora alla loro vita. In questa missione di salvataggio, ai comandi del vostro elicottero, dovete salvare questi sfortunati che precipitano prima che quattro elicotteri e due carri armati nemici possano sopprimerli.
Con il POKE apparirà la videata iniziale del gioco. Premete SELECT per decidere a che livello giocare e START per cominciare la partita. Usate il joystick in "port 1" per muovervi da destra a sinistra. Premete il pulsante FIRE per alzarvi in quota. Guadagnerete dieci punti, moltiplicati per il livello di gioco, per ogni uomo salvato. Il punteggio viene mostrato alla fine del gioco.

```
10 REM **CHOPPER MISSION
20 REM **BY Cliff McConnell
30 REM **ALL REMS MAY BE OMITTED
90 READ N0,N1,N2,N6,M,PHITE,PPOS,MPOSL
O,MPOSHI,CHITE,CPOS,DVFLAG,MSPEED,TSPE
ED,TPOS0,TPOS1,CSPEED,TIMER,SHCOLL,PA
120 TOP=PEEK(106):IVFLAG=1747:MCHAR=16
96:START=((TOP-16)*256)+230:POS=N2:DIM
 CH$(16):CH$="chopper mission "
125 DATA 0,1,2,6,0,208,209,1706,1716,1
729,1732,1741,1743,1751,1753,1754,1755
,1766,1760,256
130 DLI=START+1116:DVBLANK=START+42:IV
```

```
BLANK=START+850:DLI2=START+1144:USER=D
LI2+22
200 REM
220 MYPE=PEEK(106)-8:POKE M+N1,MYPE:PO
KE M,N0
225 POKE 106,MYPE+N2+N1:GRAPHICS 18:PO
SITION 4,N6:? #N6;"PLEASE WAIT":GOSUB
12010
230 A=MYPE*PA:FOR G=768 TO 2048:POKE A
+G,N0:NEXT G:POKE 712,224:POKE 708,62:
GOSUB 12010
240 FOR G=MPOSHI TO MPOSHI+9:POKE G,N0
:NEXT G
250 FOR G=CHITE TO CHITE+N2:POKE G,100
:NEXT G:POKE TPOS0,35:POKE TPOS1,210
260 FOR G=200 TO 207:READ N:POKE G+A+1
280,N:POKE G+A+1536,N:NEXT G:POKE 711,
26:GOSUB 12010
265 DATA 63,63,255,191,191,255,255,82
270 POKE 54279,MYPE:POKE 559,62:POKE 5
3277,3:POKE 705,120:POKE 706,204:POKE
707,46:POKE 709,70:POKE 710,212
275 GOSUB 12010
280 GOSUB 10010
310 GOSUB 11010
320 POKE IVFLAG,N1:POKE DVFLAG,N1:? #N
6;CHR$(125);:A=USR(START):POKE 1759,5:
POKE 1758,5
470 FOR G=N0 TO N2:POKE 53253+G,53+G*7
0:NEXT G:POKE 53252,123
500 REM **CHECK FOR START
505 POSITION N0,N0:? #N6;" PLEASE PRES
S START ":FOR K=N1 TO 50:IF PEEK(53279
)=N6 THEN 1020
520 NEXT K
530 POSITION N0,N0:? #N6;"score";SC;"
      ";:POSITION 11,N0:? #N6;"high";HS
;"    "
540 FOR K=N1 TO 200:IF PEEK(53279)=N6
THEN 1020
```

```
550 NEXT K:GOTO 505
1000 REM **START GAME
1020 FOR K=N0 TO 9:POKE MCHAR+K,N0:POK
E MPOSHI+K,N0:NEXT K
1030 CHOPPERS=N2+N1:SCREEN=PEEK(88)+25
6*PEEK(89):STAGE=N1:SC=N0
1035 FOR K=N0 TO 9:HI=INT((SCREEN+20+2
*K)/256):POKE 1772+K,(SCREEN+20+2*K)-H
I*256:POKE 1782+K,HI:NEXT K
1037 MLEFT=N0:POKE 1765,9
1038 FOR K=N0 TO 9:IF PEEK(1782+K)>0 T
HEN MLEFT=MLEFT+N1
1039 NEXT K:MSAVED=N0
1060 K=INT(STAGE/7):G=INT(STAGE/13):PO
KE MSPEED,STAGE*3:POKE TSPEED,4+N2*(ST
AGE-K*7):POKE CSPEED,2+STAGE-G*13
1065 IF PEEK(MSPEED)>24 THEN POKE MSPE
ED,24
1110 FUEL=1000+ABS(11-STAGE)*100
1140 POSITION N0,N0:? #N6;"          STAG
E ";STAGE;"          ";:SOUND N0,255,10,
8:SOUND 1,254,10,8
1141 FOR K=N0 TO N2:POKE CPOS+K,45:NEX
T K
1142 POKE PHITE,160:POKE PPOS,120:FOR
K=1024 TO 1280:POKE MYPE*PA+K,N0:NEXT
K:POKE 207,16
1143 POKE 53278,N0:FOR T=N0 TO N2+N2:P
OKE T+1760,N0:NEXT T
1145 ? #N6;CHR$(125);:POSITION N0,N0:?
 #N6;"score";SC:POSITION 12,N0:? #N6;"
fuel";FUEL
1146 COLOR 129:PLOT N0,11:DRAWTO 19,11
1147 FOR G=100 TO 103:POKE MYPE*PA+768
+G,252:NEXT G:POKE 53260,255
1148 FOR G=140 TO 143:POKE MYPE*PA+768
+G,3:NEXT G
1149 POKE 704,30:SOUND N1,N0,N0,N0
1150 SOUND N0,200,12,N2:A=USR(USER):GO
TO 2010+1000*PEEK(24)
```

```
2000 REM **FUEL OUT
2010 GOTO 5010
3000 REM **ENEMY HIT MAN
3010 N=0
3020 IF PEEK(53253+N)>N0 THEN 3050
3030 N=N+N1:IF N<N2+N1 THEN 3020
3040 GOTO 3091
3050 FLAG=N0:A=CPOS+N:B=CHITE+N:GOSUB
9010
3052 COL=INT(COL/2)
3055 K=PEEK(MPOSLO+COL)+PA*PEEK(MPOSHI
+COL)-SCREEN+20
3060 IF PEEK(MCHAR+COL)<>42 AND PEEK(M
POSHI+COL)<>0 THEN 3080
3070 A=TPOS0+N:GOSUB 9010:COL=INT(COL/
2)
3075 IF PEEK(MCHAR+COL)=42 THEN FLAG=N
1
3080 IF PEEK(MPOSHI+COL)=N0 THEN 3091
3081 POKE MCHAR+COL,N0:K=PEEK(MPOSLO+C
OL)+PA*PEEK(MPOSHI+COL):POKE MPOSHI+CO
L,N0:POKE K,N0:POKE MPOSLO+COL,N0
3082 POKE 1772+COL,N0:POKE 1782+COL,N0

3085 IF FLAG=N0 THEN POKE K+20,N0
3090 MLEFT=MLEFT-N1:FOR T=N0 TO 2000 S
TEP 80:SOUND N0,T,10,10:NEXT T
3091 POKE 53278,N0:FOR T=N0 TO 4:POKE
1760+T,N0:NEXT T
3092 SOUND N0,N0,N0,N0:IF MLEFT>N0 THE
N 3100
3095 IF MSAVED=N0 THEN CHOPPERS=N0:GOT
O 5030
3097 GOTO 7510
3100 GOTO 1150
4000 REM **PLAYER RESCUED MAN
4010 A=PPOS:GOSUB 9010
4015 COL=INT(COL/2):G=PEEK(MCHAR+COL)
4020 POKE MCHAR+COL,N0:K=PEEK(MPOSLO+C
OL)+PA*PEEK(MPOSHI+COL):POKE MPOSHI+CO
```

```
L,N0:POKE K,N0:POKE MPOSLO+COL,N0
4030 IF G<>42 THEN POKE K+20,N0
4040 MSAVED=MSAVED+N1:MLEFT=MLEFT-N1
4045 FOR T=255 TO N0 STEP -5:SOUND N0,
N0,N0,N0:SOUND N0,T,10,10:NEXT T
4050 IF MLEFT=N0 THEN 7510
4060 FOR G=N0 TO N2+N2:POKE SHCOLL+G,N
0:NEXT G:POKE 53278,N0
4070 GOTO 7010
5000 REM **PLAYER HIT ENEMY
5010 CHOPPERS=CHOPPERS-1
5011 POSITION N0,N0:? #N6;" CHOPPERS L
EFT:";CHOPPERS;"              "
5012 FOR T=10 TO N0 STEP -0.2:SOUND N0
,150-RND(0)*10*T,8,T:NEXT T
5020 IF CHOPPERS>0 THEN 1110
5030 POSITION N0,N0:? #N6;"       game
over      "
5035 IF SC>HS THEN HS=SC
5040 FOR T=N0 TO 400:NEXT T:GOTO 505
7000 REM
7010 SC=SC+10*STAGE:POSITION N0,N0:? #
N6;"score";SC
7020 GOTO 1150
7500 REM
7510 STAGE=STAGE+N1
7520 FUEL=(PEEK(SCREEN+16)-16)*100+(PE
EK(SCREEN+17)-16)*10+PEEK(SCREEN+18)-1
6
7530 SC=SC+INT(FUEL/10)*10
7540 POSITION 5,N0:? #N6;SC
7545 MLEFT=N0
7550 FOR G=0 TO 9:IF PEEK(1782+G)>N0 T
HEN MLEFT=MLEFT+N1
7560 NEXT G
7570 POSITION (20-MLEFT)/2,N6:FOR G=1
TO MLEFT:? #N6;"J";:NEXT G
7580 GOTO 1037
9000 REM
9010 COL=(PEEK(A)-40)/8
```

```
9020 HITE=INT((PEEK(B)-22)/16)
9030 RETURN
10000 REM **POKE IN MACHINE LANGUAGE
10010 N=START:RESTORE 32000
10015 FOR K=1 TO 4
10020 READ A:IF A()-N1 THEN POKE N,A:N
=N+N1:GOTO 10020
10025 GOSUB 12010:NEXT K
10030 K=INT(DLI/PA):G=INT(DLI2/PA):POK
E DLI2+N2,K:POKE DLI+N2,G:POKE START+2
0,G
10035 POKE DLI2+N6+N1,DLI-K*PA:POKE DL
I+N6+N1,DLI2-G*PA:POKE START+25,DLI2-G
*PA:G=INT(DVBLANK/PA)
10040 K=INT(IVBLANK/PA):POKE START+N2+
N2,G:POKE START+13,K:POKE START+N6,DVB
LANK-G*PA
10050 POKE START+15,IVBLANK-K*PA:RESTO
RE 32100:N=N0
10060 READ A:IF A()-N1 THEN POKE 1536+
N,A:N=N+N1:GOTO 10060
10070 DLIST=PEEK(560)+256*PEEK(561):PO
KE DLIST+N2+N1,PEEK(DLIST+N2+N1)+128:P
OKE DLIST+14,PEEK(DLIST+14)+128
10100 RETURN
11000 REM **POKE IN NEW CHARACTER SET
11010 CHBASE=MYPE-N2:A=CHBASE*256
11030 RESTORE 11100
11040 READ N:IF N()-N1 THEN FOR G=N0 T
O 7:READ K:POKE A+N*8+G,K:NEXT G:GOTO
11040
11045 POKE 1757,CHBASE
11050 RETURN
11100 DATA 26,24,60,24,255,24,24,36,10
2
11110 DATA 34,0,0,0,0,0,0,0,0
11120 DATA 27,0,24,60,24,255,24,24,36
11130 DATA 35,102,0,0,0,0,0,0,0
11140 DATA 28,0,0,24,60,24,255,24,24
11150 DATA 36,36,102,0,0,0,0,0,0
```

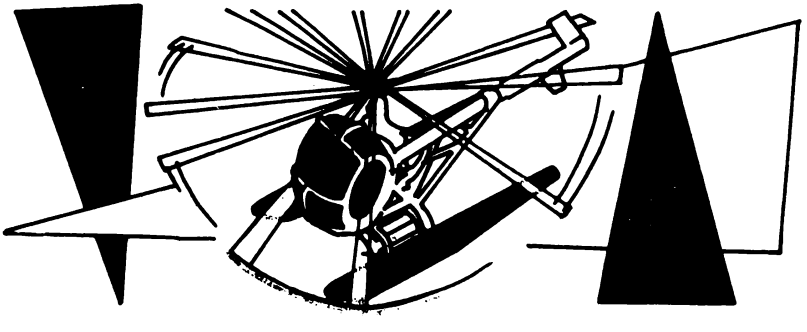
```
11160 DATA 29,0,0,0,24,60,24,255,24
11170 DATA 37,24,36,102,0,0,0,0,0
11180 DATA 30,0,0,0,0,24,60,24,255
11190 DATA 38,24,24,36,102,0,0,0,0
11200 DATA 31,0,0,0,0,0,24,60,24
11210 DATA 39,255,24,24,36,102,0,0,0
11220 DATA 32,0,0,0,0,0,0,24,60
11230 DATA 40,24,255,24,24,36,102,0,0
11240 DATA 33,0,0,0,0,0,0,0,24
11250 DATA 41,60,24,255,24,24,36,102,0

11270 DATA 42,24,60,24,255,24,24,36,10
2
11280 DATA 1,255,255,0,0,0,0,0,0
11290 DATA 0,0,0,0,0,0,0,0,0
11300 DATA -1
12000 REM **PRINT**CHOPPER MISSION
12010 FOR G=1 TO 2:POSITION POS,4:? #N
6;CH$(POS-1,POS-1);:POS=POS+N1
12020 SOUND N0,100,10,10:FOR T=N1 TO 1
0:NEXT T:SOUND N0,N0,N0,N0:NEXT G:RETU
RN
32000 DATA 104,169,7,162,128,160,42,32
,92,228,169,6,162,131,160,82,32,92,228
,169,132,141,1,2,169,120
32001 DATA 141,0,2,173,15,212,41,64,24
0,249,169,192,141,14,212,96,173,205,6,
240,3,76,98,228,165,207
32002 DATA 41,128,24,42,42,77,16,208,2
40,21,165,207,41,127,208,10,165,207,73
,128,133,207,230,207,208,21
32003 DATA 198,207,56,176,16,230,207,1
65,207,41,63,208,8,165,207,41,128,9,63
,133,207,173,16,208,208,68
32004 DATA 206,222,6,208,63,173,223,6,
141,222,6,165,88,24,105,16,133,205,165
,89,105,0,133,206,160,2
32005 DATA 177,205,24,136,113,205,136,
113,205,201,48,240,29,160,2,177,205,20
1,16,208,10,192,0,240,17,169
```

```
32006 DATA 25,145,205,208,8,56,233,1,1
45,205,24,144,3,136,16,229,165,207,41,
48,74,74,74,74,133,203
32007 DATA 165,207,41,15,24,109,206,6,
141,206,6,201,16,144,7,41,15,141,206,6
,230,203,165,207,48,9
32008 DATA 165,208,56,229,203,133,208,
208,60,173,8,208,24,109,9,208,109,10,2
08,109,11,208,41,1,208,7
32009 DATA 169,0,141,212,6,240,17,173,
212,6,208,21,173,213,6,208,7,169,1,141
,212,6,208,9,165,208
32010 DATA 24,101,203,133,208,208,10,1
65,207,16,4,169,128,133,207,165,208,20
1,42,176,8,169,0,133,207,169
32011 DATA 42,133,208,201,190,144,8,16
9,128,133,207,169,190,133,208,173,0,21
1,106,106,106,176,13,198,209,173
32012 DATA 210,6,41,2,141,210,6,24,144
,13,106,176,10,230,209,173,210,6,9,1,1
41,210,6,165,209,201,-1
32013 DATA 45,176,4,169,45,133,209,201
,200,144,4,169,200,133,209,162,2,173,2
19,6,74,74,74,133,203,173
32014 DATA 219,6,41,7,24,109,220,6,141
,220,6,201,8,144,7,41,7,141,220,6,230,
203,173,10,210,41
32015 DATA 63,208,34,173,10,210,41,15,
157,190,6,24,144,3,24,144,235,74,74,41
,1,240,6,29,199,6
32016 DATA 24,144,5,189,199,6,41,254,1
57,199,6,189,190,6,106,144,13,133,205,
189,193,6,56,229,203,157
32017 DATA 193,6,165,205,106,144,13,13
3,205,189,193,6,24,101,203,157,193,6,1
65,205,106,144,13,133,205,189
32018 DATA 196,6,24,101,203,157,196,6,
165,205,106,144,14,176,3,24,144,174,18
9,196,6,56,229,203,157,196
32019 DATA 6,189,193,6,201,42,176,13,1
```

```
89,190,6,73,1,157,190,6,169,42,157,193
,6,201,168,144,13,169
32020 DATA 168,157,193,6,189,190,6,73,
2,157,190,6,189,196,6,201,45,176,19,17
3,10,210,157,190,6,189
32021 DATA 199,6,9,1,157,199,6,169,45,
157,196,6,201,195,144,19,169,195,157,1
96,6,173,10,210,157,190
32022 DATA 6,189,199,6,41,2,157,199,6,
202,48,3,24,144,152,173,207,6,74,74,74
,74,133,203,173,207
32023 DATA 6,41,15,24,109,208,6,141,20
8,6,201,16,144,7,230,203,41,15,141,208
,6,166,203,240,50,238
32024 DATA 209,6,173,209,6,41,7,208,93
,141,209,6,160,9,185,160,6,41,63,201,4
2,240,76,185,180,6,-1
32025 DATA 240,71,132,204,133,206,185,
170,6,133,205,169,0,168,145,205,164,20
4,165,205,24,144,3,24,144,53
32026 DATA 105,20,153,170,6,165,206,10
5,0,153,180,6,185,170,6,24,105,20,133,
205,185,180,6,105,0,133
32027 DATA 206,160,0,177,205,240,12,16
4,204,185,160,6,41,192,9,42,153,160,6,
164,204,136,16,168,202,208
32028 DATA 150,173,209,6,24,105,26,133
,203,162,9,189,160,6,41,63,201,42,240,
10,189,160,6,41,192,5
32029 DATA 203,157,160,6,202,16,234,17
3,215,6,74,74,74,74,133,203,173,215,6,
24,109,216,6,141,216,6
32030 DATA 201,16,144,7,41,15,141,216,
6,230,203,173,217,6,56,229,203,141,217
,6,173,218,6,56,229,203
32031 DATA 141,218,6,173,8,208,24,109,
9,208,109,10,208,109,11,208,41,1,141,2
13,6,162,3,189,4,208
32032 DATA 157,224,6,202,16,247,173,12
,208,141,228,6,141,30,208,76,98,228,16
```

```
2,2,189,196,6,157,1,208
32033 DATA 202,16,247,173,211,6,240,3,
76,95,228,162,2,173,210,6,41,2,208,7,1
89,199,6,9,2,208
32034 DATA 5,189,199,6,41,253,157,199,
6,202,16,231,173,210,6,73,2,141,210,6,
168,169,0,192,0,240
32035 DATA 6,24,105,24,136,208,250,133
,203,170,165,1,24,105,4,133,206,169,0,
133,205,169,24,133,204,164
32036 DATA 208,189,0,6,145,205,200,232
,198,204,208,245,162,2,188,199,6,169,0
,192,0,240,6,24,105,24,-1
32037 DATA 136,208,250,157,202,6,202,1
6,235,174,202,6,169,24,133,204,172,193
,6,230,206,189,0,6,145,205
32038 DATA 200,232,198,204,208,245,174
,203,6,169,24,133,204,172,194,6,230,20
6,189,0,6,145,205,200,232,198
32039 DATA 204,208,245,174,204,6,169,2
4,133,204,172,195,6,230,206,189,0,6,14
5,205,200,232,198,204,208,245
32040 DATA 162,9,160,0,189,170,6,133,2
05,189,180,6,133,206,189,160,6,41,63,2
01,42,208,8,189,160,6
32041 DATA 145,205,24,144,26,189,160,6
,145,205,165,205,24,105,20,133,205,165
,206,105,0,133,206,189,160,6
32042 DATA 24,105,8,145,205,202,16,200
,165,209,141,0,208,238,231,6,208,3,238
,230,6,76,95,228,72,169
32043 DATA 132,141,1,2,169,120,141,0,2
,173,217,6,141,10,212,141,1,208,173,21
8,6,141,2,208,104,64
32044 DATA 72,169,132,141,1,2,169,92,1
41,0,2,173,221,6,141,10,212,141,9,212,
104,64,104,169,0,141
32045 DATA 205,6,141,211,6,165,88,24,1
05,16,133,24,165,89,105,0,133,25,160,2
,177,24,136,24,113,24
```

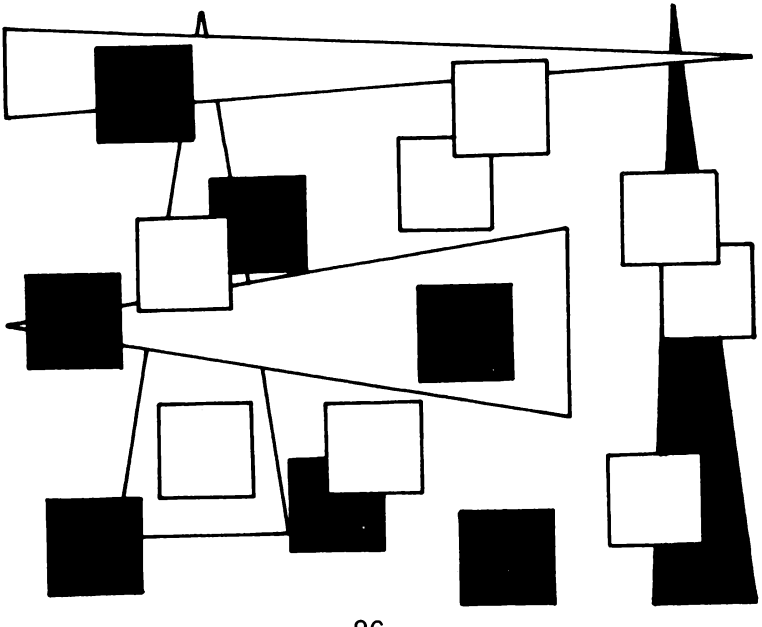
```
32046 DATA 136,113,24,201,48,208,4,169
,0,240,23,173,225,6,24,109,226,6,109,2
27,6,240,4,169,1,208
32047 DATA 7,173,224,6,240,19,169,2,23
8,205,6,172,231,6,204,231,6,240,251,23
8,211,6,133,24,96,173
32048 DATA 228,6,240,4,169,3,208,230,1
73,230,6,201,1,144,168,172,229,6,201,2
55,240,161,185,236,6,153
32049 DATA 170,6,169,0,153,160,6,185,2
46,6,153,180,6,169,0,141,230,6,206,229
,6,24,144,219,-1
32100 DATA 0,0,0,0,0,252,252,48,48,120
,120,127,255,250,250,254,32,80,0,0,0,0
,0,0,0,0
32101 DATA 0,0,0,63,63,12,12,30,30,254
,255,95,95,127,4,10,0,0,0,0,0,0,0,0,0,
0
32102 DATA 0,48,48,48,48,120,120,122,2
50,250,250,254,32,80,0,0,0,0,0,0,0,0,0
,0,0,12
32103 DATA 12,12,12,30,30,94,95,95,95,
127,4,10,0,0,0,0,0,0,0,0,255,255,255,2
55,255,255
32104 DATA 255,255,255,255,255,255,255
,255,0,0,0,0,0,0,0,-1
```



REVERSI

(CASSETTA 16K, DISCO 24K)

Il fine di questo gioco di strategia, conosciuto anche come "Otello", è di occupare la scacchiera con i propri pezzi colorati in azzurro. Voi ed il computer giocate a turno piazzando un pezzo per volta sulla scacchiera sino a che non rimangano caselle libere. Per "mangiare" i pezzi del computer e farli diventare del vostro colore, dovete muovere in modo che due dei vostri pezzi circondino uno o più pezzi del computer; questo può avvenire verticalmente, orizzontalmente o diagonalmente.
Premete SELECT per decidere chi inizia e START per cominciare la partita. Sullo schermo apparirà la scacchiera insieme al vostro cursore: adoperate un joystick in "port 1" per portare il cursore sulla casella desiderata e premete il pulsante FIRE per collocare il pezzo.



```
0 REM ******** REVERSI *********
1 REM *** by G. Ryan, 1984 ***
10 N0=0:N1=1:GOSUB 6000:DIM A(10,10)
20 FOR A=N1 TO 10:FOR B=N1 TO 10:IF A(
>N1 AND B<>N1 AND A<>10 AND B<>10 THEN
 A(A,B)=161:NEXT B:NEXT A
30 A(A,B)=131:NEXT B:NEXT A
40 CP=4:HP=36:SD=131:BL=161:HX=5:HY=5
50 A(5,5)=CP:A(6,6)=CP:A(5,6)=HP:A(6,5
)=HP:P=0:R=33
150 GOSUB 3000:IF W=N0 THEN GOTO 2000
999 REM COMPUTERS TURN
1000 POKE 77,N0:POSITION 1,11:? #6;"
          (MY TURN!"
1010 S=HP:T=CP:H=0
1020 FOR A=2 TO 9:FOR B=2 TO 9:IF A(A,
B)<>BL THEN 1170
1030 Q=0:FOR C=-1 TO 1:FOR D=-1 TO 1:K
=0:F=A:G=B
1040 IF A(F+C,G+D)<>S THEN 1060
1050 K=K+1:F=F+C:G=G+D:GOTO 1040
1060 IF A(F+C,G+D)<>T THEN 1080
1070 Q=Q+K
1080 NEXT D:NEXT C
1090 IF A=2 OR A=9 THEN Q=Q*2
1100 IF B=2 OR B=9 THEN Q=Q*2
1110 IF A=3 OR A=8 THEN Q=Q/2
1120 IF B=3 OR B=8 THEN Q=Q/2
1130 IF (A=2 OR A=9) AND (B=3 OR B=8)
THEN Q=Q/2
1140 IF (A=3 OR A=8) AND (B=2 OR B=9)
THEN Q=Q/2
1150 IF Q<H OR Q=0 OR (RND(0)>0.3 AND
Q=H) THEN 1170
1160 H=Q:M=A:N=B:HX=A-1:HY=B-1
1170 NEXT B:NEXT A
1180 IF H=0 AND R=0 THEN 5000
1190 IF H=0 THEN 1210
1200 GOSUB 4000
1210 Z=TP:GOSUB 3000:IF Z=TP THEN 5000
```

```
1999 REM PLAYERS TURN
2000 POKE 77,N0:POSITION 1,11:? #6;"!Y
OUR TURN!            ":GOSUB 8500
2010 GOSUB 8000:IF L=0 THEN GOSUB 8500
:GOTO 2010
2020 S=CP:T=HP
2030 GOSUB 4000
2040 GOSUB 3000
2050 GOTO 1000
2999 REM ADD UP PIECES
3000 C=0:H=0
3010 FOR B=2 TO 9:FOR A=2 TO 9:POSITIO
N 4+A,B-1:? #6;CHR$(A(A,B));:C=C+(A(A,
B)=CP):H=H+(A(A,B)=HP):TP=C+H
3020 NEXT A:NEXT B
3030 POSITION 9,10:? #6;H;" ":POSITION
 18,10:? #6;C;" "
3040 IF C+H>=64 THEN POP :GOTO 5000
3050 RETURN
3999 REM PLACE PIECE ON BOARD
4000 FOR C=-1 TO 1:FOR D=-1 TO 1:F=M:G
=N
4010 IF A(F+C,G+D)<>S THEN 4030
4020 F=F+C:G=G+D:GOTO 4010
4030 IF A(F+C,G+D)<>T THEN 4060
4040 A(F,G)=T:IF M=F AND N=G THEN 4060

4050 F=F-C:G=G-D:GOTO 4040
4060 NEXT D:NEXT C
4070 RETURN
4999 REM GAME OVER
5000 H=0:C=0:FOR B=2 TO 9:FOR A=2 TO 9
:H=H+(A(B,A)=HP):C=C+(A(B,A)=CP)
5010 NEXT A:NEXT B
5020 POSITION 9,10:? #6;H;" ";:POSITIO
N 18,10:? #6;C;" ";
5030 X=INT(H/(C+H)*100+0.5):POSITION 1
,11:? #6;" your pieces ";X;"%";"    ";
5040 POSITION 1,2:? #6;"GAME":POSITION
 1,4:? #6;"OVER"
```

```
5050 POSITION 15,2:? #6;"press":POSITI
ON 15,4:? #6;"start"
5060 IF PEEK(53279)<>6 THEN 5060
5070 PF=1:GOSUB 6100:GOTO 20
5999 REM INITIALIZATION
6000 RT=PEEK(740)-4:POKE 106,RT-2:GRAP
HICS 18:POKE 756,RT:RT=RT*256:DIM SD$(
10),BD$(10)
6010 POSITION 7,4:? #6;"reversi":POSIT
ION 1,6:? #6;"WRITTEN BY G. RYAN":FOR
A=1 TO 10:SD$(A)=CHR$(131)
6020 BD$(A)=CHR$(161):NEXT A:FOR A=RT
TO RT+7:POKE A,0:NEXT A
6030 FOR A=40 TO 47:POKE RT+A,PEEK(573
44+A):NEXT A
6040 FOR A=112 TO 511:POKE RT+A,PEEK(5
7344+A):NEXT A:RESTORE 6060
6050 READ I:IF I<>-1 THEN FOR A=I TO I
+7:READ D:POKE RT+A,D:NEXT A:GOTO 6050

6060 DATA 32,0,60,126,126,126,126,60,0

6070 DATA 8,0,126,126,126,126,126,126,
0
6080 DATA 24,170,85,170,85,170,85,170,
85
6090 DATA 72,255,129,129,129,129,129,1
29,255
6100 DATA 96,255,195,129,129,129,129,1
95,255,-1
6110 PUT #6,125:POSITION 5,0:? #6;SD$:
POKE 708,126:POKE 709,56:POKE 710,164:
POKE 711,70
6120 POSITION 5,9:? #6;SD$:FOR A=1 TO
8:POSITION 5,A:? #6;SD$(10);BD$(3);SD$
(10):NEXT A
6130 POSITION 1,10:? #6;"you own 0":PO
SITION 12,10:? #6;"; own 0"
6140 POSITION 1,11:? #6;"select:COMP G
O 1ST";:W=1
```

```
6150 IF PEEK(53279)=5 THEN 6180
6160 IF PEEK(53279)=6 THEN 6220
6170 GOTO 6150
6180 W=(W=0):IF W THEN POSITION 8,11:?
 #6;"COMP GO 1ST"::GOTO 6200
6190 POSITION 8,11:? #6;" YOU GO 1ST";
6200 IF PEEK(53279)<>7 THEN 6200
6210 GOTO 6150
6220 POSITION 1,11:? #6;"
     ";:RETURN
7999 REM INPUT PLAYERS MOVE WITH JOYST
ICK
8000 LOCATE 5+HX,HY,CH:HH=HX:HV=HY
8010 COLOR CH+8:PLOT 5+HX,HY:ST=STICK(
0):IF STRIG(0)=0 THEN 8050
8015 IF PEEK(53279)=1 THEN POP :COLOR
CH:PLOT HX+5,HY:GOTO 5000
8020 HX=HX+PTRIG(0)-PTRIG(1):HY=HY+(ST
=13 OR ST=5 OR ST=9)-(ST=14 OR ST=10 O
R ST=6)
8030 HX=HX+(HX<1)-(HX>8):HY=HY+(HY<1)-
(HY>8):IF HX<>HH OR HY<>HV THEN COLOR
CH:PLOT 5+HH,HV:GOTO 8000
8040 GOTO 8010
8050 IF CH<>161 THEN GOSUB 8500:GOTO 8
010
8060 COLOR CH:PLOT 5+HH,HV
8070 M=HX+1:N=HY+1:L=0:X=HX+5:Y=HY
8080 FOR A=-1 TO 1:FOR B=-1 TO 1
8090 LOCATE X+A,Y+B,C:IF C=4 THEN 8150
8100 NEXT B:NEXT A:RETURN
8150 H=X+A+A:V=Y+B+B
8160 LOCATE H,V,Z:IF Z=HP THEN L=N1:PO
P :POP :RETURN
8170 IF Z=131 OR Z=161 THEN 8100
8180 H=H+A:V=V+B:GOTO 8160
8500 SOUND N0,15,12,4:FOR Z=N1 TO 20:N
EXT Z
8510 IF STRIG(N0)=N0 THEN 8510
8520 SOUND N0,N0,N0,N0:RETURN
```

MARS LANDER

(CASSETTA 16K, DISCO 24K)

Orbitando intorno a Marte improvvisamente vi accorgete di aver poco carburante a disposizione. Poiché il carburante è insufficiente per sfuggire all'attrazione gravitazionale del pianeta, dovrete tentare un difficile atterraggio su una vecchia piattaforma posta sulla superficie di Marte. Prima di toccare il suolo dovete evitare asteroidi di ogni dimensione, particelle di polvere stellare ed un campo di forza rotante. La gravità farà sentire sempre più la sua forza e, per evitare un impatto violento, dovete consumare prezioso carburante.
Premete START per iniziare il gioco. Usate il joystick in "port 1" per muovere l'astronave a destra e a sinistra. Spostate in basso il joystick per accelerare la velocità di caduta e premete il pulsante FIRE per fornire una spinta verso l'alto alla vostra nave in modo da contrastare la forza di gravità.

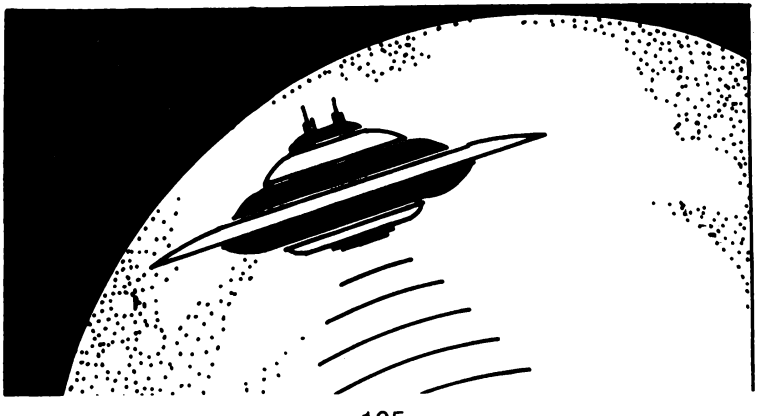
```
10 REM **MARS LANDER
20 REM **By Cliff McConnell
30 REM **ALL REMS MAY BE OMITTED
120 POKE 704,0
130 HS=0:GOSUB 10010:GOSUB 12010
140 GOSUB 15010:? #6;CHR$(125):GOSUB 1
2010:GOSUB 13010:GOSUB 14010:Y=IN+37:X
=124:STAGE=1
143 POKE 53760,150:ND=0:L=3:SC=0:POKE
656,0:POKE 657,7:? "      ";
144 POKE 53270,0:W=20-STAGE*2:GOSUB 16
010:IF W<8 THEN W=8
145 COLOR 1:PLOT 0,50:DRAWTO 159,50:SH
=20:COLOR 0:PLOT 30,50:DRAWTO 30-W,50
148 Y=IN+37:X=124:U=35-RND(0)*10:G=30:
PM$(1)=CHR$(0):PM$(2304)=CHR$(0):PM$(2
)=PM$:F=500
```

```
149 POKE 53760,150:POKE 656,0:POKE 704
,250:TRAP 710:POKE 53278,0
150 V=STICK(0):G=G+V(V):Y=Y+G(G):U=U+H
(V):X=X+G(U):PM$(Y)=SH$:POKE 53248,X:F
=F+F(V):POKE 657,17:? F;" ";
155 COLOR 0:PLOT SH,50:COLOR 1:PLOT SH
-W,50:SH=SH+1
160 POKE 53761,N(V):IF F AND  NOT PEEK
(53252) THEN 150
165 POKE 53761,0:POKE 77,0
170 V=PEEK(53252):IF V>4 THEN 210
180 IF V>3 THEN 2010:REM BONUS
190 IF V>1 AND G<35 THEN 1010:REM LAND
ED
200 REM **DIED
210 PM$(Y)=EXPL$:FOR T=0 TO 255 STEP 5
:SOUND 0,T,10,10:POKE 704,(255-T)/16:N
EXT T
220 FOR T=255 TO 0 STEP -7:SOUND 0,T,1
0,10:NEXT T:SOUND 0,0,0,0
230 L=L-1:POKE 656,1:POKE 657,18:? L;
240 IF L>0 THEN 148
510 IF SC>HS THEN HS=SC
515 GOTO 140
520 RUN
710 COLOR 1:PLOT (SH-W)*(SH>W),50:SH=(
SH+1)*(SH<159+W):TRAP 710:GOTO 160
1000 REM **LANDED
1010 POP :STAGE=STAGE+1:COLOR 0:IF STA
GE/3=INT(STAGE/3) THEN COLOR 1
1020 PLOT 25,62:DRAWTO 135,62:POKE 656
,0
1045 POKE 656,1:POKE 657,7:? STAGE;
1050 SOUND 0,121,10,8:SOUND 1,120,10,8
:FOR K=0 TO 30:NEXT K:SOUND 0,0,0,0:SO
UND 1,0,0,0:FOR K=0 TO 5:NEXT K
1060 SOUND 0,121,10,8:SOUND 1,120,10,8
:FOR K=1 TO 200:NEXT K:SOUND 0,0,0,0:S
OUND 1,0,0,0:POKE 656,0
1080 FOR K=1 TO F STEP 5:F=(F-5)*(F-5)
```

```
0):SC=SC+1:POKE 657,7:? INT(SC);:POKE
657,17:? F;" ";:SOUND 0,100,10,10
1090 FOR T=0 TO 10:NEXT T:SOUND 0,0,0,
0:NEXT K:SC=INT(SC)
1950 GOTO 144
2000 REM **GOT BONUS
2010 FOR T=0 TO NO
2011 IF  NOT BY(T) THEN NEXT T
2013 COLOR 0:PLOT BX(T),BY(T):POKE 532
78,0
2020 FOR K=200 TO 0 STEP -10:SOUND 0,K
,10,10:NEXT K:POKE 53760,150:V=PEEK(53
252)
2030 IF V<4 THEN SC=SC+STAGE*10:POKE 6
56,0:POKE 657,7:? SC;:BX(T)=0:BY(T)=0:
GOTO 150
2040 COLOR 3:PLOT BX(T),BY(T):NEXT T:G
OTO 150
10000 REM **SET UP PM GRAPHICS
10010 GRAPHICS 7:DL=PEEK(560)+256*PEEK
(561):POKE DL+85,70:POKE DL+88,6:POKE
DL+89,6:POKE DL+90,6:POKE 752,1
10030 Y=0:X=0:DIM PM$(2304),SH$(11),V(
15),H(15),G(80),F(15),N(15),BX(11),BY(
11),EXPL$(11)
10130 A=ADR(PM$):PM=INT(A/2048)*2048:X
=PM+1024:IF A>X THEN PM=PM+2048:X=X+20
48
10140 IN=X-A:POKE 710,158:POKE 708,52
10150 POKE 54279,PM/256:POKE 559,62:PO
KE 53277,3:POKE 53248,150
10200 FOR K=1 TO 11:READ V:SH$(K)=CHR$
(V):NEXT K
10210 DATA 0,0,0,16,16,56,124,40,0,0,0

10220 FOR K=0 TO 15:READ V:N(K)=V:NEXT
 K
10230 DATA 0,0,0,0,0,4,134,4,0,4,134,4
,0,0,134,0
10240 FOR K=1 TO 11:READ V:EXPL$(K,K)=
```

```
CHR$(V):NEXT K
10250 DATA 0,73,42,28,8,127,8,28,42,73
,0
11000 REM **JOYSTICK VARIABLES
11010 RESTORE 11020:FOR K=0 TO 15:READ
 V,H:V(K)=V:H(K)=H:NEXT K
11020 DATA 0,0,0,0,0,0,0,0,0,0,.4,.2,-
.3,.2,.2,.2,0,0,.2,-.2,-.3,-.2,.2,-.2,
0,0,.4,0,-.3,0,.4,0
11030 FOR K=0 TO 60:G(K)=(K-30)*0.1:NE
XT K:FOR K=61 TO 80:G(K)=3:NEXT K
11040 FOR K=0 TO 15:F(K)=0:NEXT K:F(14
)=-1:F(10)=-1:F(6)=-1
11050 RETURN
12000 REM **DRAW SCREEN
12010 COLOR 1:PLOT 0,0:DRAWTO 159,0:DR
AWTO 159,79:DRAWTO 0,79:DRAWTO 0,0
12015 PLOT 0,79:T=0
12020 X=INT(RND(0)*5):T=T+INT(RND(0)*5
):IF T<159.5 THEN DRAWTO T,79-X:GOTO 1
2020
12990 RETURN
13000 REM **SET UP OBSTACLES
13010 FOR T=0 TO 5:PLOT RND(0)*158,RND
(0)*12+52:NEXT T
13020 COLOR 2:PLOT 77,74:DRAWTO 83,74
13030 COLOR 1:PLOT 84,70:DRAWTO 84,75:
DRAWTO 76,75:DRAWTO 76,70
13050 RETURN
14000 REM **PRINT SCORE AND FUEL
14010 SOUND 0,0,0,0
14015 POKE 656,0:POKE 657,21:? "HIGH S
CORE:";HS;
14020 POKE 657,1:? "SCORE:";SC;"     ";
:POKE 657,12:? "FUEL:";F;
14050 POKE 656,1:POKE 657,1:? "STAGE:1
          ";:POKE 657,12:? "SHIPS:3";

14220 RETURN
15000 REM **WAIT FOR START
```

```
15010 POKE 656,1:POKE 657,1:? "please
press start";
15015 IF HS>0 THEN POKE 656,0:POKE 657
,32:? HS;
15018 POKE 656,1:POKE 657,21:? "MARS L
ANDER";
15020 IF PEEK(53279)<>6 AND STRIG(0)<>
0 THEN 15020
15030 RETURN
16000 REM **PUT UP EXTRA OBSTACLES
16010 COLOR 1:FOR T=1 TO 3:PLOT RND(0)
*158,RND(0)*12+52:NEXT T
16020 N=RND(0)*4:FOR T=1 TO N:S=INT(RN
D(0)*3)+1:H=RND(0)*(157-10*S)+3*S:V=RN
D(0)*(39-7*S)+10
16030 PLOT H+5*S,V:RESTORE 16200:FOR K
=1 TO 23:READ X,Y:DRAWTO H+X*S,V+Y*S
16035 NEXT K:NEXT T
16040 COLOR 0:FOR T=0 TO NO:PLOT BX(T)
,BY(T):NEXT T
16050 COLOR 3:NO=NO+(NO<10)*1:FOR T=0
TO NO:BX=RND(0)*158:BY=RND(0)*40+20:BX
(T)=BX:BY(T)=BY:PLOT BX,BY:NEXT T
16080 RETURN
16200 DATA 3,0,3,1,2,1,2,2,1,2,1,4,0,4
,0,5,2,5,2,5,2,6,4,6,4,7,5,7,5,6,6,6,6
,4,7,4,7,3,6,3,6,2,5,2,5,0
```



Come scrivere programmi migliori

Impiegare il proprio tempo battendo i programmi come quelli che vi abbiamo presentato va benissimo, ma verrà certo il momento in cui deciderete di mettere a punto dei programmi di gioco personali. In questa sezione del libro vorrei quindi discutere alcune idee che potranno esservi d'aiuto nella realizzazione dei vostri 'games', rendendo piacevole la fase d'elaborazione e — cosa ancora più importante — divertente il momento del gioco.

CHIAREZZA D'IDEE

Vi accorgerete che in molti casi (per non dire sempre) il programma che state elaborando per il vostro computer comincia a vivere di vita propria già durante la stesura, deviando progressivamente dall'idea che vi eravate prefissi all'inizio della programmazione. Ciò nonostante nelle fasi preliminari di lavoro è importante avere un'idea chiara dei vari elementi che entreranno in gioco.
Questo mio consiglio è meno banale di quanto ptrebbe sembrare. Naturalmente se state pensando ad un gioco in cui « inghiottite alcune pillole iperricostituenti e dovete inseguire dei fantasmi attraverso un labirinto », sapete di non poter usare lo schema di programmazione valido, ad esempio, per un gioco che vi trasporta all'interno di una Quercia Incantata dove vivono elfi e folletti. In ogni caso dovete superare lo stadio elementare dell'« Ora mi scrivo un'avventura » per elaborare invece una serie di punti come: (a) l'oggetto del gioco; (b) l'effetto visivo che si vuole ottenere; (c) le variabili ed i nomi variabili necessari; (d) il tipo di input per il giocatore; (e) le modalità per la derminazione della vittoria e della sconfitta, eccetera.
Esaminiamo queste voci una ad una.

OGGETTO DEL GIOCO

Di norma è possibile sintetizzare questo punto in poche parole: « Trovare il tesoro degli Aztechi », « Distruggere il maggior numero possibile di asteroidi prima di esaurire le navi spaziali » o « Giocare una partita a scacchi ». Pur essendo di rapida esecuzione, questa fase iniziale nella produzione del gioco non deve essere tralasciata. Vi consiglio, anzi, di mettere per iscritto la vostra formula — che può consistere in una sola frase, ma che potrebbe anche raggiungere e superare il paragrafo se fosse necessario mettere a punto più "quadri" con scenari differenti.
Certamente potete accantonare il vostro obbiettivo originale se il corso preso dal programma durante l'elaborazione vi sembra più soddisfacente. Indipendentemente da questo fatto è importante che abbiate qualcosa di concreto cui mirare, se non volete perdere ore ed ore gingillandovi inutilmente con il computer.

VISUALIZZAZIONE

L'esperienza personale mi insegna che è estremamente utile fare qualche schizzo del tipo di visualizzazione grafica che si vuole ottenere a programma inserito. Una volta che avete preparato il disegno (e non importa quanto sia approssimativo purché evidenzi tutti gli elementi che volete avere sul video, con le relative posizioni e dimensioni), l'idea alla base del vostro programma si cristallizzerà più facilmente. Non solo capirete immediatamente come scrivere certe parti del codice in modo da conseguire l'obbiettivo del gioco, ma potrete anche valutare l'opportunità o meno di realizzare il programma nella forma in cui lo avevate ideato. Lasciando sullo schermo tutti gli elementi che vi proponevate, il gioco potrebbe forse risultate troppo complicato, in altri casi potrebbe accadere che il video appaia in gran parte vuoto e che dobbiate quindi escogitare uno scenario di gioco più complesso.
Vi posso assicurare che prima di iniziare la programmazione vi tornerà utile schematizzare l'effetto visivo finale, e specialmente se state ideando 'arcade games' o giochi

di simulazione. Otterrete così delle indicazioni relative alle variabili da adottare, ai grafici definibili dall'utente e al miglior tipo di input da proporre ai giocatori per conseguire una buona interazione tra partners.
Come probabilmente sapete, nei giochi di simulazione il computer riproduce una realtà esterna — ad esempio la conduzione di un'impresa, di una guerra o di un aeroporto — consentendovi di provare (approssimativamente) le esperienze che affrontereste partecipando realmente ad un'impresa del genere. I giochi di simulazione non presentano grandi difficoltà di stesura — non di codificazione, almeno — ma richiedono una metodica e scrupolosa impostazione del programma.
In un mio libro, « The ZX Spectrum Explored » (Sinclair Browne, 1982), è contenuto un programma dal titolo piuttosto inverosimile: « Lavorare per l'Uomo ». Il giocatore è posto a dirigere un'industria la cui forza lavoro, notevolmente variabile, viene impiegata per fabbricare un favoloso prodotto detto 'Zibby'. Due o tre volte a settimana al dirigente viene sottoposta una relazione industriale in base alla quale egli (o ella) deve stabilire quanti operai assumere o licenziare (se ci riesce), la quantità di Zibbies che costituisce l'obbiettivo di produzione della settimana, eccetera.
Questa relazione è la chiave del programma e per definire il gioco ho iniziato col tracciare uno schizzo degli elementi da visualizzare. La mia bozza era all'incirca la seguente:

RELAZIONE INDUSTRIALE: SETTIMANA 5

Il capitale liquido ammonta a $ 2657,92
I magazzini contengono 12 Zibbies per un valore di $ 169,68
Ciascuno viene venduto a $ 14,14 e costa al produttore $ 7,41
La forza lavoro è costituita da 7 persone
Il salario individuale è di $ 41
Questa settimana le spese salariali ammontano a $ 287
Ogni operaio può fabbricare 10 Zibbies alla settimana, per una produzione totale di 70 Zibbies.
Delineato questo schema potevo andare avanti. Come vedete dalla bozza si possono ricavare utili indicazioni circa le variabili da adottare. Tanto per cominciare sapevo

di dover controllare il numero della settimana, il capitale, il contenuto dei magazzini (e il suo valore) e così via.
Ho potuto sperimentare che una volta completato lo schema il resto della programmazione non presentava particolari difficoltà.
Uno schizzo eseguito secondo queste indicazioni vi offrirà una guida immediata alle variabili di cui dovrete servirvi.

VARIABILI UTILI

Generalmente tendo ad usare delle variabili che abbiano una certa attinenza con i vari elementi del gioco; con questo procedimento, infatti, evito di fare una lista delle variabili adottate e dei loro oggetti di riferimento. Per esempio potrei usare SA per settimana, CL per capitale liquido, PZ per il costo di produzione di ogni Zibby e VZ per il suo prezzo di vendita. Così, posto che il numero di Zibbies corrisponda a Z, potrei calcolare il valore globale della merce in mio possesso moltiplicando Z (numero di Zibbies) per VZ (prezzo di vendita), e il totale dei costi di produzione moltiplicando Z per PZ (costo di produzione). Naturalmente se vendessi tutti gli Zibbies di cui dispongo il mio profitto sarebbe dato da Z×VZ - Z×FZ. Scoprirete che seguendo questo tipo di logica è molto più facile tenersi costantemente aggiornati sugli sviluppi del programma.

INPUT DEL GIOCATORE

L'essenziale è scrivere giochi divertenti e facili da eseguire. Non conta nulla disporre del miglior programma che sia mai stato ideato con gli asteroidi, se poi il gioco risulta difficoltoso perché avete sistemato il pulsante per aprire il fuoco proprio accanto al tasto di 'rotazione'.
Per i comandi 'su', 'giù', 'destra' e 'sinistra', in molti programmi si adottano automaticamente i tasti di controllo cursore o quelli contrassegnati dalle frecce, senza considerare che per il giocatore è estremamente scomodo servirsene. Date uno sguardo alla vostra tastiera e cercatene di migliori. Per i programmi che consentono solo spostamenti verso destra e sinistra io adotto spesso

la 'M' e la 'Z', utilizzando la barra spaziatrice per far fuoco. L'uso di questi tasti mi sembra logico, non si perde tempo per impararli ed è estremamente semplice ricordarli quando il gioco è in atto. Allo stesso modo tendo ad usare 'A' (su) e 'Z' (giù) per la mano sinistra ed i tasti 'maggiore di' e 'minore di' per la destra e la sinistra (facendo rilevare ai giocatori che i simboli < e > muovono nelle direzioni in cui sono rispettivamente rivolti)
Quando è possibile usate INKEY$ o GET$, evitando così ai giocatori di adoperare i tasti RETURN o ENTER per avviare il programma.

CONCLUSIONI DEL GIOCO

È necessario definire e illustrare chiaramente ai giocatori le modalità che regolano la vittoria e la sconfitta.
Per vincere dovete far saltare in aria tutti gli alieni. Perdete automaticamente se un extraterrestre atterra mentre disponete ancora di qualche navicella spaziale o, semplicemente, quando avete esaurito le astronavi. Nei giochi a due, esce sconfitto chi. per primo perde sette pezzi o viene ucciso per tre volte, in altri casi il gioco si conclude solo quando la differenza tra i due punteggi è tre, sette o un qualsiasi altro numero che vi aggradi.
Elaborate queste disposizioni ed esponetele con linearità ai giocatori. Si tratti di liberare la parte sinistra del video da hippies in protesta o di registrare la bella somma di 7,3 miliardi di dollari, lo scopo del gioco deve essere sempre chiaro allo sfidante e possibile da conseguire. Ai livelli più alti diventa veramente frustrante trovarsi di fronte a delle « condizioni di vittoria » irraggiungibili. Non fate che frodare i giocatori se ponete degli obbiettivi che i vari impedimenti inseriti nel gioco rendono impossibile conseguire; e non importa quante difficoltà incontriate elaborando il programma.
Spero che questi cinque punti vi suggeriscano qualche idea per continuare a scrivere programmi che, pur essendo di semplice realizzazione, possano risvegliare in voi e nei vostri amici il piacere del gioco.

Glossario

A

Accumulatore — Il posto all'interno del computer entro il quale si compiono calcoli aritmetici e dove i risultati di tali calcoli sono immagazzinati.
Algoritmo — La serie di passi compiuti dal computer per risolvere un particolare problema.
Alfanumerico — Questo termine è generalmente usato in relazione ad una tastiera. Esempio: "è una tastiera alfanumerica". Ciò significa che la tastiera ha sia lettere sia numeri. Indica anche "la serie di caratteri" del computer e comprende i numeri e le lettere che il computer può riprodurre sul video.
ALU (Arithmetic/Logic Unit) — La parte del computer adibita ad operazioni aritmetiche (come l'addizione e la sottrazione) e in cui vengono prese le decisioni.
AND — Un'operazione logica "booleana" che il computer usa nei suoi processi decisionali. È basata sull'algebra di "Boole", un sistema sviluppato dal matematico George Boole (1815-64). Nell'algebra di Boole le variabili di un'espressione sono soggette alle operazioni logiche come AND, OR, NOR, ecc.
ASCII — Sta per American Standard Code for Information Exchange (Codice Americano Standard per lo Scambio di Informazioni) ed è il sistema in codice più largamente usato per la lingua inglese con caratteri alfanumerici. Ci sono 128 lettere maiuscole e minuscole, cifre e alcuni caratteri speciali. Il codice ASCII trasforma i simboli e le istruzioni di controllo in combinazioni binarie a sette "bit".
ASSEMBLER — Un programma che trasforma i programmi scritti nel linguaggio Assembly in codice macchina (che il computer può comprendere in modo diretto). Il linguaggio Assembler è un linguaggio di programmazione a basso livello che usa le combinazioni facilmente memorizzabili di due o tre lettere per rappresentare una particolare istruzione che l'Assembler poi trasforma affinché il computer possa comprenderlo. Esempio: ADD (aggiungere) e SUB (sottrarre). Un computer programmatico nel linguaggio Assem-

bler tende a lavorare più velocemente di un altro programmato in un linguaggio di alto livello come il BASIC.

B

BASIC —È un acronimo per Beginners All-Purpose Symbolic Instruction Code. È il linguaggio per computer più largamente usato nel settore dei micro-computer. Sebbene sia stato molto criticato, ha il vantaggio di essere molto facile da imparare. Molte espressioni BASIC sono simili all'inglese comune.
BAUD — Viene da Baudot, un pioniere della comunicazione telegrafica. Baud misura il livello di trasmissione delle informazioni ed è approssimativamente uguale a un bit per secondo.
BCD — Un'abbreviazione per Binary Coded Decimal (Binario Codificato Decimale).
Benchmark — Un test con cui possono essere misurate certe funzioni del computer. Ci sono molti "test standard Benchmark", che generalmente sperimentano soltanto la velocità. Questo è un aspetto raramente importante in un microcomputer, il tipo di computer piú interessante per un potenziale compratore.
Binario — Un sistema numerico che usa soltanto i numeri zero e uno.
Bit — Un'abbreviazione per Binary Digit (cifra binaria). È la piú piccola unità di informazione che un circuito di computer può riconoscere.
Boole, Algebra di— Il sistema algebrico sviluppato dal matematico George Boole che usa numerazioni algebriche per esprimere relazioni logiche (vedi AND).
Bootstrap — Un breve programma o sottoprogramma che viene letto all'interno del computer quando è acceso per la prima volta. Orienta il computer ad accettare i programmi successivi, di lunghezza maggiore.
Bug — Un errore che impedisce il funzionamento del programma. Sebbene sia generalmente usato per indicare solo un difetto o un errore nel programma, il termine bug può anche essere usato per un difetto nell'hardware del computer.

Bus — Un numero di conduttori usati per trasmettere segnali, dati e istruzioni.
Byts — Un gruppo di cifre binarie che compone una parola di computer. Generalmente i bits presenti in un byte sono otto.

C

CAI — Computer Assisted Instruction
CAL — Computer Assisted Learning. Il termine è generalmente usato per descrivere quei programmi che coinvolgono lo studente in processi di apprendimento.
CHIP — Il termine che si usa per indicare l'intero circuito, che è inciso su un piccolo pezzo di silicio. Il chip è, naturalmente, il cuore del computer.
CLOCK — Il congegno di sincronizzazione all'interno del computer che sincronizza le sue operazioni.
COBOL — Un linguaggio ad alto livello che deriva dalle parole Common Business Orientated Language. Il COBOL è usato principalmente per la schedatura e il mantenimento di ciò che è stato registrato.
Comparator — Un congegno che mette a confronto due cose e produce un segnale collegato con la differenza tra i due.
Compiler — Un programma che trasforma i linguaggi di programmazione ad alto livello in codici binari da macchina. In questo modo i programmi scritti in linguaggi ad alto livello possono essere utilizzati dal computer.
Complement — Un numero che è derivato da un altro secondo regole stabilite.
Computer — Un congegno con tre principali capacità o funzioni:
1) accettare i dati
2) risolvere i problemi
3) fornire risultati.
CPU — Sta per Central Processing Unit. È il cuore dell'intelligenza del computer dove si gestiscono i dati e si sviluppano le istruzioni.
Cursore — Un carattere che appare sul video quando il computer sta compiendo le sue operazioni. Esso rivela do-

ve il prossimo carattere sarà stampato. Su un computer ci sono generalmente "tasti di controllo cursore" per permettere all'utente di spostare il cursore sul video.

D

Dati — Informazioni presentate in una forma che il computer può elaborare.
Debug — Il termine che si usa quando si esamina un programma e si correggono eventuali errori, cioè, trovare e rimuovere i bugs, i difetti.
Digital Computer — Un computer che opera su quelle informazioni che si presentano come digitali.
Disk/Disc — È un disco di plastica sensibilizzato magneticamente, un poco più piccolo di un "45 giri". È usato per l'immagazzinamento dei programmi e per ottenere dati. I dischi sono notevolmente più veloci da caricare rispetto ad una cassetta della stessa lunghezza di programma. Si può accedere molto velocemente su un disco mentre un programma sta operando per ottenere ulteriori dati.
Display — L'output visivo del computer, generalmente su un video, o su uno schermo di monitor.
Dot Matrix Printer — Una stampante che stampa i listati di un programma o di ciò che appare sul video. Ogni lettera e ogni carattere sono composti da un certo numero di "dots", cioè di punti. Più alto è il numero di punti per ogni carattere e migliore sarà la qualità operativa della stampante.
Dynamic Memory — Un'unità di memoria all'interno del computer che "dimentica" ciò che contiene quando viene tolta l'alimentazione elettrica.

E

Editor — Questo termine è generalmente usato per designare quella sezione del computer che permette al programmatore di cambiare le istruzioni di un programma mentre lo sta scrivendo.

EPROM — Sta per Erasable Programmable Read-Only Memory. È come il ROM nel computer, con la sola differenza che è abbastanza facile inserire materiale all'interno di una EPROM che non sparisce quando si toglie l'alimentazione. Le EPROM devono essere esposte a forti raggi ultravioletti se si vuole cancellarle.
Error Messages (Messaggi di errore) — L'informazione data da un computer che indica dove è stato commesso un errore nella codificazione durante una parte del programma. L'informazione è trasmessa dal computer che si ferma e stampa una parola, o una parola e dei numeri, o soltanto una combinazione di numeri, in fondo al video. Questo rivela quale errore è stato fatto. Gli errori più comuni includono l'uso della lettera O invece dello zero in una linea, o l'omissione in una espressione di entrambe o di una delle parentesi, o l'errore nella definizione di una variabile.

F

File — Una serie di item d'informazione collegati fra loro e organizzati in modo sistematico.
Floppy Disk — Un disco magnetico relativamente poco costoso, usato per immagazzinare le informazioni del computer, e così chiamato perché molto flessibile (vedi Disk/Disc).
Flow Chart — Un diagramma tracciato prima di scrivere un programma nel quale lé principali operazioni sono racchiuse entro rettangoli o altre forme e connesse a frecce attraverso linee per rappresentare "loop" di istruzioni, e le decisioni scritte fra parentesi. Ciò ti aiuta a scrivere un programma in modo molto più semplice perché trappole come loop infiniti o variabili non definite possono essere scoperte in ogni fase. Può risultare utile scrivere questo diagramma per programmi molto brevi, ma è senz'altro conveniente se si vuole creare un programma più lungo.
Firmware — Ci sono tre tipi di "ware" nei computer: cioè programmi software "temporanei"; hardware, come quelli contenuti nelle ROM permanenti; e firmware, nel quale l'informazione è relativamente permanente, come in una EPROM (vedi EPROM).

Flip-Flop — Un circuito che mantiene in memoria una condizione elettrica finché questa non è cambiata nella condizione opposta da un segnale.
FORTRAN — Sta per FORmula TRANslation (traduzione di formula). È un linguaggio di computer ad alto livello, orientato verso problemi matematico-scientifici.

G

Gate — Un circuito elettrico che, sebbene possa captare uno o più segnali in arrivo, manda in uscita soltanto un singolo segnale.
Graphics — Informazione grafica, in opposizione alle informazioni fornite da lettere e numeri.

H

Hard Copy — Uscita di computer il cui supporto è permanente.
Hardware — Le parti fisiche di un computer (vedi anche software e firmware).
Hexadecimal (Hex) — Un sistema numerico con base sedici, Sono usate le cifre da zero a nove e le lettere A, B, C, D, E, F per la rappresentazione dei numeri. A è uguale a 10, B è uguale a 11, C è uguale a 12, e così via. Hex è spesso usato dagli utenti di microcomputer.
Hex Pad — Una tastiera specificamente progettata per inserire numerazioni esadecimali.
High Level Language (Linguaggio ad alto livello) — Un linguaggio di programmazione che permette all'utente di parlare con il computer più o meno in lingua inglese. In generale più è alto il livello del linguaggio (cioè, più è simile all'inglese) più lungo sarà il tempo impiegato dal computer per tradurlo in un linguaggio che esso può utilizzare. Linguaggi a più basso livello sono molto più difficili per l'operatore umano ma generalmente offrono una esecuzione più veloce.

I

Input — L'informazione inserita nel computer attraverso una tastiera, un microfono, una cassetta o un disco.
Input/Output (I/O Device) — Un congegno che accetta le informazioni o le istruzioni dal mondo esterno, le trasmette al computer e, dopo l'elaborazione, le rinvia, o sotto una forma adattabile alla memorizzazione, o una forma comprensibile all'essere umano.
Instruction — Il dato che comanda una sola azione nell'elaborazione delle informazioni operate dal computer (noto anche come comando).
Integrated Circuit (Circuito Integrato) — Un completo circuito elettronico residente sulla superficie di un semiconduttore.
Interface — Il confine fra il computer e un periferico come la stampante.
Interpreter — Un programma che traduce, istruzione per istruzione, il linguaggio ad alto livello inserito da un operatore umano in un linguaggio che la macchina può capire.
Inverter (Invertitore) — Un "gate" logico che cambia nell'opposto il segnale inserito.
Interactive Routine (Sottoprogramma Interattivo)— Parte di un programma che è ripetuto più volte finché non si raggiunge una data condizione.

J

Jump Instruction — Un'istruzione che dice al computer di muoversi verso un'altra parte del programma, quando la destinazione di questo spostamento dipende dal risultato di un calcolo appena compiuto.

K

K — Questa lettera riporta la misura della memoria. La memoria è generalmente misurata in blocchi di "K". Un K contiene 1.024 bytes.

Keyword (parola-chiave) — La parola di inizio in una linea di programmazione, generalmente la prima parola dopo il numero della istruzione. Parole-chiavi sono STOP, PRINT e GOTO.

L

Language — I linguaggi di computer sono divisi in tre gruppi: i linguaggi ad alto livello, come il BASIC, che sono relativamente vicini all'inglese ed abbastanza facili da usare per l'uomo; i linguaggi a basso livello, come l'ASSEMBLER, in cui compaiono brevi frasi che hanno qualche collegamento con l'inglese (ADD per add e RET per return, ad esempio); e il codice macchina, che comunica più o meno direttamente con la macchina.
LCD — Sta per Liquid Crystal Diode. Alcuni computer come il TRS-80 Pocket Computer usano un "display" LCD.
LED — Sta per Light Emitting Diode. I numeri luminosi rossi che sono spesso usati negli orologi da polso o da muro sono composti da LED.
Logic — La formula matematica di uno studio di relazioni fra eventi.
Loop — Una serie di istruzioni all'interno di un programma che sono ripetute finché una particolare condizione viene soddisfatta.

M

Machine Language o Machine Code (Codice macchina) — Un codice che può essere capito e messo in pratica direttamente dal computer.
Magnetic Disk— vedi Disk e Floppy Disk.
Mainframe — I computer sono generalmente divisi in tre gruppi e il fatto di appartenere ad un certo gruppo dipende dalla grandezza del computer. Il computer più venduto è il microcomputer. I computer di media grandezza sono i minicomputer, e i computer giganti che qualche volta si vedono nei film di fantascienza sono computer "mainframe". Fino a 15 anni fa tali computer erano, in termini pratici, i so-

li disponibili.
Memory — Dentro un computer ci sono due tipi di memoria. La prima chiamata ROM, è la memoria che arriva già programmata sul computer e che dice al computer come prendere decisioni e come compiere operazioni aritmetiche. Questa memoria non è cancellata se si spegne il computer. Il secondo tipo è la RAM. Questa memoria mantiene il programma che è stato scritto sulla tastiera o che è trasmesso all'interno tramite una cassetta o un disco. La maggior parte dei computer "dimenticano" ciò che è in RAM quando vengono spenti.
Microprocessor — Il cuore di qualsiasi computer. Richiede le interfacce con le unità periferiche, l'alimentazione di energia e i congegni di input e output. In tal modo può operare come un microcomputer.
MODEM — Sta per Modulatore/Demodulatore. È un apparato che permette a due computer di parlare fra loro per telefono. Generalmente i computer usano un supporto nel quale è posto un ricevitore telefonico.
Monitor — Nel linguaggio dei computer ha due significati. Un significato è uno schermo simile a quello televisivo. Un monitor ha serie difficoltà ad adattarsi ai programmi televisivi e generalmente l'immagine prodotta su un monitor è migliore di quella prodotta da una comune Tv. Il secondo significato di un monitor si rapporta alla ROM. Il monitor di un computer è descritto come il complesso di informazioni incorporate nel computer all'atto dell'acquisto. Queste informazioni permettono di prendere decisioni e di compiere calcoli aritmetici.
Motherboard — Una struttura alla quale possono essere aggiunti circuiti extra. Questi circuiti spesso offrono facilitazioni che non sono incorporate nel computer, come quella di produrre suoni o di controllare una penna ottica.
MPU — Abbreviazione per Microprocessor Unit.

N

Nano-secondo — Un nano-secondo è un millimiliardesimo di secondo, l'unità di tempo con la quale si misura la velocità di un computer o di un microcircuito di memoria.
Non-Volatile Memory — La memoria che non si perde quando il computer è spento. Alcuni computer più piccoli

come il TRS-80 Pocket Computer hanno "non volatile memory". Le batterie mantengono il programma inserito per settecento ore.

Not - Un'operazione booleana che trasforma una cifra binaria nel suo opposto.

Null String — Una "stringa" che non contiene caratteri. Nel programma compare sotto forma di due doppie virgolette, senza niente fra di loro.

Numerico — Concerne i numeri quando sono opposti alle lettere (cioè alfabetico). Molte tastiere sono alfanumeriche, sono cioè provviste sia di numeri sia di lettere.

O

Octal — Un sistema numerico che usa otto come base e quindi le cifre da 0 a 7. Tale sistema non è oggi molto usato nel settore dei microcomputer. Il sistema esadecimale è più comune (vedi Hexadecimale)

Operating System — (Sistema operativo) — Il software e il firmware, generalmente previsti su una macchina che permette di far girare altri programmi.

OR — Un'operazione booleana che ritorna a 1 se uno o più input sono 1.

Oracle (Oracolo) — Un metodo di messaggi invianti test tramite un segnale di trasmissione televisiva. Un set di teletest è richiesto per decodificare i messaggi.

Output — Informazioni o dati trasmessi dal computer a congegni quali uno schermo come quello televisivo, una stampante o una cassetta. L'output generalmente consiste in un'informazione che il computer ha prodotto come risultato della elaborazione di un programma.

Overflow — Un numero troppo grande o troppo piccolo per essere elaborato dal computer.

P

Pad — Vedi Keypad

Pagina — Spesso usata per indicare la quantità di informazioni necessaria per riempire uno schermo televisivo. Così, vedendo una pagina del programma, è possibile analizzare la quantità di informazioni che appaiono sul video tutte in una volta.
PASCAL — Un linguaggio ad alto livello.
Periferico — Qualsiasi cosa che è collegata e controllata dal computer, come un'unità a disco, una stampante o un sintetizzatore vocale.
Port — Un connettore attraverso il quale le informazioni sono trasmesse o inserite nel computer.
Prestel — Il nome inglese per un sistema basato sulla trasmissione di informazioni via telefono da un computer centrale e sulla loro visualizzazione su uno schermo televisivo. Negli Stati Uniti una versione commerciale simile è nota come "The Source".
Program — Nel linguaggio computer può essere una lista di istruzioni che si inseriscono nel computer, oppure può essere un verbo, cioè "programmare un computer".
PROM — Sta per Programmable Read Only Memory. È un sistema che può essere programmato e, una volta che lo è stato, il programma è permanente (vedi anche EPROM e ROM).

R

Random Access Memory (RAM) — La zona di memoria entro il computer che può essere cambiata a comando dalla persona che usa il computer. Il contenuto della RAM è di solito perduto quando un computer è spento. La RAM è l'integrato che memorizza ciò che viene scritto e anche i risultati di calcoli in atto.
Read-Only Memory (ROM) — In contrasto alla RAM, l'informazione qui non può essere cambiata dall'utente e non va perduta quando si spegne il computer. I dati della ROM sono collocati dal produttore e dicono al computer il modo con cui deve prendere decisioni e come compiere calcoli aritmetici. La misura di capacità della ROM e RAM è data in unità K (vedi K).
Recursion — La ripetizione continua di una parte del programma.

Registro — Una specifica sezione della memoria nella quale uno o più parole di computer sono memorizzate nel corso delle operazioni.
Parola Riservata — Una parola che non può essere usata per una variabile in un programma perché il computer la leggerà in modo errato. Un esempio è la parola TO. Poiché TO ha uno specifico significato nel linguaggio dei computer, la maggior parte dei calcolatori respinge questa parola come nome per una variabile. Lo stesso vale per parole come FOR, GOTO e STOP.
Routine — Questa parola può essere usata come sinonimo di programma o può riferirsi a una specifica sezione all'interno del programma (vedi anche Subroutine).

S

Seconda Generazione — Ha due significati. Il primo si applica nei confronti dei computer che usano transistor, in opposizione alla prima generazione di computer che usavano valvole. La seconda generazione può anche indicare la seconda copia di un particolare programma. Susseguenti generazioni sono danneggiate da un disturbo crescente.
Semiconduttore — Un materiale che è generalmente un isolante elettrico, ma sotto specifiche condizioni può diventare un conduttore.
Serial — Informazione che è memorizzata o inviata in una sequenza, un bit alla volta.
Segnale — Un impulso elettrico che trasmette dati.
Silicon Valley — Il nome popolare dato a una zona in California dove si trovano molti produttori di semiconduttori.
SNOBOL — Un linguaggio ad alto livello.
Software — Il programma inserito nel computer dall'utente. Questo programma dice al computer cosa fare.
Software Compatible — Si riferisce a due differenti computer che possono accettare i programmi scritti per l'altro.
Static Memory — Un congegno di memoria non volatile che trattiene le informazioni per tutto il tempo che il computer è acceso. Tuttavia, non richiede ulteriori consumi di energia per mantenere in ordine la memoria.
Subroutine (Sottoprogramma) — Parte di un programma che è spesso inserita molte volte durante l'esecuzione del programma principale. Una subroutine finisce con un'istru-

zione che comanda di ritornare indietro alla istruzione successiva a quella che aveva inviato lo svolgimento del programma alla subroutine.

T

Teletext — Informazione trasmessa nella sezione superiore di un'immagine che appartiene a una trasmissione televisiva. Richiede una struttura speciale per essere decodificata e per riempire il video di informazioni riguardo il test. I messaggi teletext possono anche essere trasmessi tramite un cavo, per esempio, il servizio Prestel in Gran Bretagna o The Sources negli Stati Uniti.
Teletype — Un apparecchio simile ad una macchina da scrivere che può mandare, ricevere e stampare informazioni.
Terminale — Un'unità indipendente dell'unità centrale di elaborazione. Generalmente consiste di una tastiera e di un visore.
Time Sharing — Un processo attraverso il quale molti utenti possono aver accesso a un grande computer che si sposta rapidamente da un utente all'altro in sequenza, cosicché ogni utente ha l'impressione di essere il solo utente del computer.
Truth Table (Tavola della verità) — Una tavola matematica che elenca tutti i possibili risultati di un'operazione booleana. I risultati dipendono dalle varie combinazioni di input.

U

UHF — Ultra High Frequency (300-3000 MegaHertz).
Ultra Violet Erasing — La luce ultravioletta deve essere usata per cancellare le EPROM (vedi EPROM).

V

Variabile — Una lettera o combinazione di lettere e simbo-

li a cui il computer può assegnare un valore o una parola durante il funzionamento di un programma.
VDU — Abbreviazione per Visual Display Unit.
Volatile — Indica la memoria che "dimentica" le informazioni in essa contenute quando il computer è spento.

W

Word (Parola) — Un gruppo di caratteri o una serie di cifre binarie che rappresentano un'unità d'informazione e occupano una singola posizione di memoria. Il computer elabora una parola come singola istruzione.
Word-Processor (Elaboratore di testi) — Una macchina da scrivere altamente intelligente che permette a chi scrive di manipolare il testo, di spostarlo, per giustificare margini e per spostare interi paragrafi, se necessario, su un video prima di mandare l'informazione sulla stampante. Questi elaboratori hanno generalmente memorie, cosicché modelli di lettere e testi di lettere, scritti precedentemente, possono essere nuovamente stampati.

Traduzioni

TRAFFIC PANIC

150 — Traffico pericoloso
4010 — Premi start/select
4015 — Traffico pericoloso
4020 — Punti
4022 — Punteggio massimo
4025 — Livello
4070 — Seleziona
7115 — Stadio
9030 — Tempo
13110 — Fine gioco

BREAKTHROUGH

130 — Punti
140 — Premi option/select
150 — Numero palle
420 — Punteggio massimo
425 — Dimostrazione
440 — Fine gioco - Premi start
450 — Gioco escluso
4000 — Veloce
4010 — Piano

SHOOTOUT

50 — Fine
5190 — Angolo - Angolo
5200 — Sacchi - Sacchi

TOWER OF FLAMES

450 — Attendere prego
5060 — Punti
5070 — Punteggio massimo
5090 — Livello
5110 — Premi start/select
7740 — Rimangono
8920 — Stadio
17110 — Rimangono
19010 — Fine gioco

ASTEROID STORM

115 — Tempesta di asteroidi - Prego attendere
650 — Premi start per favore
660 — Punti - Punteggio massimo
690 — Punti
710 — Stadio
1515 — Fine gioco

BLOCKOUT

670 — Premi start, prego - Punteggio massimo
710 — Option: 1 giocatore - Select: 5 turni
770 — 2 giocatori
780 — 1 giocatore
800 — Turni
930 — Fine gioco - premi start

SPACE ATTACK

220 — Nome
240 — Attacco spaziale - I migliori dieci
310 — Prepararsi all'attacco
550 — Ondata di attacco - Terminata
610 — Fine gioco - Punti
630 — Punteggio massimo
680 — Grande punteggio!
690 — Inserisci il tuo nome - nella tabella dei record
850 — Missione fallita
860 — Fine gioco
960 — Punti - Ondate - Vite
980 — Astronave extra!

GOBBLER

515 — Punti - Massimo
520 — Premi start/select
525 — Stadio - seleziona
528 — Select
610 — Inizio
626 — Stadio
645 — Punti - Stadio
1030 — Tacchini rimasti
2030 — Punti
4120 — Punti

GRAND PRIX

90 — Stadio
120 — Fine
150 — Corsa - Terminate - Incidenti
160 — Livello di abilità - Massimo livello di abilità
170 — Premi start per cominciare un'altra corsa
240 — Fine gioco
260 — Livello di abilità - Massimo livello di abilità - Premi start
665 — Premi start

ESCAPE

230 — Fine gioco
250 — Punteggio - Premi start
640 — Punti - Stadio

TIME TYPIST

210 — Punti
220 — Punteggio massimo
230 — Premi start, per favore
510 — Tempo
8010 — Che sfortuna
10010 — Preparati

GALAX ATTACK

130 — Attacco galattico
5010 — Attendere prego
5110 — Ci manca poco
5210 — Quasi finito
5411 — Massimo punteggio
5412 — Premi start per cominciare il gioco
5414 — Seleziona il livello
5416 — Premi option per le bombe
11075 — Stadio
15010 — Astronavi rimaste
15060 — Fine gioco

CHOPPER MISSION

225 — Prego attendere
505 — Premi start per favore
530 — Punti - Massimo punteggio
1140 — Stadio
1145 — Punti - Carburante
5011 — Elicotteri rimasti
5030 — Fine gioco
7010 — Punti

REVERSI

2000 — Tocca a te!
5030 — Le tue pedine
5040 — Fine gioco
5050 — Premi start
6130 — Tu hai - Io ho
6140 — Seleziona: inizia il computer
6180 — Inizia il computer
6190 — Inizi tu

MARS LANDER

14015 — Punteggio massimo
14020 — Punti - Carburante
15050 — Stadio: 1 - Navi: 3
15010 — Premi start, per favore

Computer Games

GIOCHI PER IL VOSTRO ATARI

Tanti fantastici programmi, con la traduzione in italiano, appositamente ideati per questa collana e in grado di garantirvi ore e ore di svago istruttivo e divertente

Tra i giochi spettacolari di questo libro: ATTACCO DALLO SPAZIO (difendete la vostra stazione spaziale dagli invasori alieni rompiscatole); MISSIONE SPECIALE (siete alla guida dell'elicottero di soccorso: il destino degli ostaggi dipende da voi); BREAKTHROUGH (misurate la vostra capacità di coordinazione e i vostri nervi saldi in questa veloce versione del "Breakout"); STORMO DI ASTEROIDI (guidate con cura la vostra fidata astronave, siete in un mare di rocce vaganti).

"ALTRI GIOCHI PER IL VOSTRO ATARI" vi aiuterà moltissimo, giocando, a migliorare la vostra abilità fornendovi non solo tutte le istruzioni per inserire correttamente i programmi nel computer, ma anche un utilissimo glossario dei termini essenziali e preziosi consigli e indicazioni su come modificare e migliorare i programmi del libro o realizzarne di nuovi.

Tanti fantastici giochi per tutta la famiglia!

CL 006-0153-5 ISBN 88-7605-153-8

L. 9.500 (i.i.)

Altri giochi per il vostro ATARI